DOCUMENT RESUME

ED 318 443 IR 014 353

TITLE	LOGO K-12.
INSTITUTION	New York City Board of Education, Brooklyn, N.Y.
REPORT NO	ISBN-0-88315-795-0
PUB DATE	86
NOTE	443p.
AVAILABLE FROM	New York City Board of Education, Curriculum Unit, P.S. 206, Room 310, Neck Road and East 22nd Street, Brooklyn, NY 11229.
PUB TYPE	Guides - Classroom Use - Guides (For Teachers) (052)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Computer Graphics; *Computer Literacy; *Computer Science Education; Curriculum Guides; Early Childhood Education; Elementary Secondary Education; Glossaries; *Microcomputers; *Programing; Programing Languages; Special Education
IDENTIFIERS	*LOGO Programing Language

ABSTRACT

This curriculum guide presents lessons in the use of the LOGO programing language for students at all grade levels. The first section contains 19 lessons for the early childhood level, covering topics such as how computers work, directionality and moving the turtle, and creating designs. The 76 lessons for grades K-6 that are described in the next section cover such topics as the history and uses of computers, keyboarding, following directions, moving the turtle, LOGO commands, editing in LOGO, creating designs, and creating and moving sprites. The third section presents 70 lessons for grades 7-12 on topics such as applications of computers, word processing, LOGO commands, recursive programming, using variables, creating designs, LOGO programs, writing with LOGO, LOGO sounds and music, random procedures, and function machines. Performance objectives, vocabulary, motivation, aim, development, application, summary, and exercises are included for most lessons. The appendixes include a discussion of computers and special education students; and a glossary of computer terms. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED318443

IR014353

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

LOGO
K-12

·LOGO
√45
412.511
LOGO

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Dorothy Klein

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

NEW YORK CITY
BOARD OF EDUCATION

OFFICE OF TECHNOLOGY
AND DIVISION OF
CURRICULUM AND INSTRUCTION

BEST COPY AVAILABLE

LOGO
K-12

LOGO
√4.5
412.511
LOGO

NEW YORK CITY
BOARD OF EDUCATION

OFFICE OF TECHNOLOGY
AND DIVISION OF
CURRICULUM AND INSTRUCTION

New York City Board of Education

Robert F. Wagner, Jr., *President*

Irene Impellizzeri, *Vice President*

Gwendolyn C. Baker
Richard I. Beattie
Stephen R. Franse
James F. Regan
Edward L. Sadowsky

Nathan Quinones, *Chancellor*

Charles I. Schonhaut, *Deputy Chancellor*

Jerald Posman, *Deputy Chancellor for Financial Affairs*

Louise Latty, *Chief Executive for Instruction*

Charlotte Frank, *Executive Director*
Division of Curriculum and Instruction

Irwin Kaufman, *Director*
Office of Technology

Copyright (c) 1986
by the Board of Education of the City of New York

Application for permission to reprint any section of this material should be made to the Chancellor, 110 Livingston Street, Brooklyn, N.Y. 11201. Reprint of any section of this material shall carry the line, "Reprinted from *Logo K-12* by permission of the Board of Education of the City of New York."

New York City Public Schools should order additional copies of this publication from the Bureau of Supplies.

ISBN No. 0-88315-795-0 Curric. No. 00-7000-70

Other persons and institutions may obtain copies of this publication from the Curriculum Production Unit of the Office of Curriculum Development and Support, Room 613, 131 Livingston Street, Brooklyn, N.Y. 11201 For information call (718) 596-4902.

Foreword

In 1983 the Division of Curriculum and Instruction published computer literacy curriculum guides for both elementary and secondary schools. That project was the first step in a comprehensive plan for computer education in New York City. Computer technology is new and it is changing daily. We as educators are faced with a common challenge: how to use this exciting and powerful technology in a way that will meet the changing needs of the students and will continue to accommodate changes as they occur. The challenge is a great one, but the potential for these tools is even greater.

We know that computers can ignite the imagination and extend the limits of human learning. They enable their users to view the world in new ways. When students interact with computers they take charge of their own learning and learn more through this interaction. In short, computers are tools used by humans for human endeavors.

Logo K - 12 is an important step in the New York City plan for computer education. Logo is a sophisticated and flexible computer language. As students gain proficiency in Logo, they develop important concepts in computer science. The content of this curriculum will challenge students to master the reasoning skills demanded for programming Logo. We hope that this guide will aid teachers and students in keeping pace with the challenges of our information age.

Charlotte Frank

Charlotte Frank
Executive Director
Division of Curriculum and Instruction

Acknowledgments

LOGO K-12 is a joint project of the Office of Technology, Irwin Kaufman, Director and the Division of Curriculum and Instruction, Charlotte Frank, Executive Director.

Joseph Eaione, Assistant Director, Office of Technology, gave impetus and guidance to this project by arranging for meeting facilities and resources and providing overall supervision.

Stephen Radin, Citywide Curriculum Administrator, Office of Technology, served as principal author and editor for this project.

A team of dedicated professionals contributed extensively to the original draft, including:

Nadine Baca Scala, C.S.D. 2, Orlando Barzaga, P.S. 144 M., Gwen Brown, C.S.D. 3, Patrick J. Camerlengo, I.S. 61 R., Frances Curcio, Queens College, Cathi Dean, P.S. 144M., Elaine Fitzpatrick, I.S. 61 R., Debra Freedman, I.S. 88K., Sharon Funk, P.S. 251 K., Fred Goldberg, Manhattan Technical Assistance Center, Richard Gordon, C.S.D. 19, Marion Hollingsworth, C.I.S. 229 X., Patrick Ianniello, P.S. 177 K., Harriet Isecke, Hackensack Schools, Annette Jackson, C.E.S. 229 X., Coran James, Manhattan Technical Assisance Center, Amy Kargman, C.S.D. 18, D. Lynn Randle, P.S. 18 R., Larry Rosen, I.S. 44 M., Elayne Schulman, Staten Island Academy, A.A. Violenus, P.S. 112 M. Ruth Weber, C.S.D. 21, Sandra Welsher, Office of Technology, and Stephen Wulfson, C.I.S. 174 X.

The following reviewers made helpful comments and suggestions:

Maryann Marrapodi-Guzman, Marc Licht, Jean Ruchalla and Sidney Schwager.

This manuscript was prepared for publication by the Office of Technology, under the direction of Stephen Radin.

Word processing: Norma Quinones, Erin Lamb
Graphics/Production: Ernest Mortuzans
Cover: Victor Roseman

Introduction

The use of computers in the classroom has become more prevalent during the past few years. Through this use we have developed a knowledge of their functions and a better vision of their potential. We encourage our colleagues to freely experiment and attempt new approaches. Only with all of us working in concert, can we fulfill our vision for an effective integration of school technology into all subject areas. From these beginnings, we will nurture the critical thinking and practical skills our students will need to succeed in the information age. If children are not comfortable with computers, they will as be unprepared for the future as an adult who cannot read or write.

The purpose of this guide is to give teachers and supervisors a working knowledge of various approaches to enhance pupil learning. Our guides are intended to be open ended and flexible in nature. As a 1920 car is not efficient by 1980 standards, outmoded computer education ideas of the seventies have been replaced by the new visions of the eighties. Classroom use of technology has doubled and redoubled in the past few years. Teachers are continually devising new ideas for the use of this technology.

Our primary responsibility in this area and our reason for writing this book, is to see that computers are used as an efficient classroom tool. Only through a real sharing can we move forward in this rapidly changing field of technology. Feel free to write to me directly with your comments and suggestions concerning this curriculum guide, so that we may improve and update it to meet your needs.

Irwin Kaufman

Irwin Kaufman
Director
Office of Technology

It is the policy of the New York City Board of Education not to discriminate on the basis of race, creed, national origin, age, handicapping condition, or sex in its educational programs, activities, and employment policies, as required by law. Inquiries regarding compliance with appropriate laws may be directed to Mercedes Nesfield, Director, Office of Equal Opportunity, 110 Livingston Street, Brooklyn, N.Y. 11201; or to Charles Tejeda, Director, Office of Civil Rights, Department of Education, 26 Federal Plaza, Room 33-130, New York, New York 10278

Table of Contents

NOTES ON THE USE OF THIS GUIDE

What is LOGO? xiii

EARLY CHILDHOOD TOPICS

Note to Teachers 3
Beyond Keystrokes 4
Introduction To Early Childhood Computer Experience 5
The Computer System 5
The Computer 7
How Computers Work 8
Playing Turtle 10
Directionality: Movements Forward and Back 11
Directionality: Movements Left And Right 14
Moving the Turtle Forward, Back, Right and Left 17
Practice with Forward, Back, Right and Left 19
Creating Designs Using Forward, Back, Right and Left 20
Introducing Geometric Shapes 21
Creating Line Designs on the Computer 23
Creating Squares on the Computer 25
Creating Rectangles on the Computer 26
Creating Designs with Squares and Rectangles 28
Adding Color to Turtle Designs 29
Changing Background Colors 30

ELEMENTARY GRADE TOPICS (K - 6)

Topic 1 What is a Computer? 33
Topic 2 Computers at Home 36
Topic 3 Computers in School 38
Topic 4 Careers with Computers 40
Topic 5-E Computers of the Past - Who Developed the First "Computers"? 42
Topic 5-A What were the First "Real" Computers Like? 46
Topic 6-A Modern Computers 48
Topic 7 The Computer as a Problem-Solving Tool 50
Topic 8-A Computer Ethics - What's Right? 52
Topic 9-A Advantages to Business and Government 55
Topic 10-A Advantages to Organized Sports and the Creative Arts 58
Topic 11-A Advantages in Education, Medicine and Law 61
Topic 12-A Computer Disadvantages - Privacy and Criminality 64
Topic 13-A Computers in the News 67
Topic 14 Computer Etiquette - Do's and Don'ts 70
Topic 15-A Computer Futures 72

Topic 16-A Types of Software 76
Topic 17 Following Directions 81
Topic 18 Practice Keyboarding Skills on the Apple 84
Topic 19 Practice Keyboarding Skills on the Commodore Pet 87
Topic 20 Practice Keyboarding Skills on the TRS-80 90
Topic 21-E Verbal Direction Game 93
Topic 21-M Verbal Direction Game 95
Topic 21-A Verbal Direction Game 97
Topic 22-E Playing Turtle 99
Topic 22-A Playing Turtle 101
Topic 23-E Editing Directions in a Verbal Game 104
Topic 23-A Walking a Maze 105
Topic 24 Importance of Correct Sequence 107
Topic 25 Geometric Shapes 110
Topic 26-E Introduction to LOGO 111
Topic 26-A Introduction to LOGO 113
Topic 27-E Logo Lines 116
Topic 27-M Logo Lines 119
Topic 27-A Logo Lines 122
Topic 28-E PENUP and PENDOWN Commands 125
Topic 28-A PENUP and PENDOWN Commands 127
Topic 29 Hide The Turtle 129
Topic 30-E The Repeat Command 131
Topic 30-A The Repeat Command 134
Topic 31 A Logo Procedure 138
Topic 32 Procedures Inside of Procedures 140
Topic 33-E Editing in LOGO 142
Topic 33-A Editing in LOGO 147
Topic 34-E Squares 151
Topic 34-M Squares 153
Topic 35-E Triangles 156
Topic 35-A Triangles 159
Topic 36-E Rectangles 163
Topic 36-A Rectangles 167
Topic 38-E Colors 170
Topic 38-A Colors 173
Topic 39 "Walking Geometric Shapes" Like a Turtle 175
Topic 40-E Polygons and Their Origins 178
Topic 40-A Polygons and Their Origins 181
Topic 41 Polygons 183
Topic 42-E Using Variables 185
Topic 43-M Variables and Polygons 188
Topic 44-E Circles 190
Topic 44-A Circles 192
Topic 45 Coordinates 195
Topic 46-E LOGO and Words (for Apple Terrapin Logo) 198
Topic 46-A LOGO and Words (for Apple Terrapin Logo) 201
Topic 47 Writing with LOGO (for Apple Terrapin Logo) 204
Topic 48 Naming (for Apple Terrapin Logo) 206
Topic 49 LOGO Conversations (for Apple Terrapin Logo) 208

Topic 50-E Introduction to Sprites 211
Topic 50-A Introduction to Sprites 213
Topic 51-E Making Sprites Move 215
Topic 52-A Making Sprites Move 217
Topic 53-E Giving Sprites Direction 219
Topic 53-A Giving Sprites Direction 221
Topic 54-E Many Sprites at Once 223
Topic 54-A Sprites and More Sprites 224
Topic 55-E Making New Shapes 225

SECONDARY GRADE TOPICS (7 - 12)

Topic 1 Computers in Society 229
Topic 2 Future Computers 231
Topic 3 The Computer as a Problem-Solving Tool 233
Topic 4 The Effect of Computer Design on Problem-Solving 235
Topic 5 Computer Disadvantages - Job Security 239
Topic 6 Disadvantages of Computers-Depersonalization and Dependency . 242
Topic 7 Software Storage and Management 244
Topic 8 Introduction to the Keyboard 248
Topic 9 The Micro as a Word Processor 252
Topic 10 Knowledge of a Word Processor 255
Topic 11 Logo Closed Figure 258
Topic 12 Logo Closed Figure and Procedures 260
Topic 13 Edit Mode 262
Topic 14 PENUP and PENDOWN Commands 268
Topic 15-E Recursive Programming 271
Topic 15-M Recursive Programming 274
Topic 15-A Recursive Programming 277
Topic 15-E Using Variables 280
Topic 16-A Using Variables 282
Topic 17-E Using Several Variables 285
Topic 17-M Using Several Variables 287
Topic 17-A Using Several Variables 289
Topic 18 Hide the Turtle 291
Topic 19-A Squares 293
Topic 20-A Triangles 297
Topic 21-A Rectangles 300
Topic 22-A Circles 303
Topic 23-A Colors 307
Topic 24-A Polygons 310
Topic 25-E Programs in LOGO 312
Topic 25-M Programs in LOGO 314
Topic 25-A Programs in LOGO 316
Topic 26-A Coordinates 318
Topic 27-A LOGO and Words (for Apple Terrapin Logo) 322
Topic 28-M Writing with Logo (for Apple Terrapin Logo) 325
Topic 28-A Writing with Logo (for Apple Terrapin Logo) 328

Topic 29-M Naming (for Apple Terrapin Logo) 330
Topic 29-A Naming (for Apple Terrapin Logo) 332
Topic 30-M LOGO Poems (for Apple Terrapin Logo) 336
Topic 30-A LOGO Poems (for Apple Terrapin Logo) 339
Topic 31-M LOGO Conversations (for Apple Terrapin Logo) 342
Topic 31-A LOGO Conversations (for Apple Terrapin Logo) 346
Topic 32-E The Wait Command 349
Topic 32-M The Wait Command 351
Topic 33-M Making New Shapes 353
Topic 34-E Magnifying a Shape 355
Topic 34-M Magnifying a Shape 358
Topic 35-E Total Turtle Trip Setheading 360
Topic 35-M Total Turtle Trip Setheading 363
Topic 36-E Setheading with a Variable 367
Topic 36-M Setheading with a Variable 370
Topic 37-E Setheading with Conditionals 375
Topic 37-A Setheading with Conditionals 378
Topic 38-E Logo Sounds and Music 380
Topic 38-M Logo Sounds and Music 383
Topic 38-A Logo Sounds and Music 386
Topic 39-E Music 390
Topic 39-M Music 392
Topic 39-A Music 395
Topic 40-E Random Procedures 398
Topic 40-M Random Procedures 400
Topic 40-A Random Procedures 402
Topic 41-E Output 404
Topic 41-M Output 406
Topic 41-A Output 408
Topic 42-E Tiles and Characters 410
Topic 42-M Tiles and Characters 415
Topic 42-A Tiles and Characters 417
Topic 43 Advanced Topic: Passing Values 421
Topic 44 Advanced Topic: Approaching Infinity 423

APPENDICES

A - Computers and Special Education Students 426
B - Glossary 429

NOTES
ON THE USE
OF THIS GUIDE

What is Logo?

You may have heard Logo talked about as: a philosophy, a culture, and a computer language. How can Logo be all these things?

To help understand how Logo encompasses a wide range of areas, we have collected some of the thoughts and experiences of the great minds who developed Logo and those of many of the teachers in the field -- all whom are excited by the promise afforded by this exciting, new, learning environment.

WHAT IS THIS PROMISE?

Logo offers a new way for a student to "think about thinking", to explore powerful ideas and to fulfill self-initiated goals. Teachers of Logo have observed exciting new interactions among students and between students and teachers. Cognitive development and exploration in problem solving skills have been noted in addition to the discovery of powerful mathematical ideas.

WHY DO WE BELIEVE LOGO OFFERS LEARNERS, BOTH STUDENTS AND TEACHERS, THE BEST TOOL FOR LEARNING?

Logo encourages the development of "self work" and provides a risk free environment for exploration. Developing Logo projects through exploration and discovery is a cornerstone of the Logo philosophy. As the nature of knowledge changes learning strategies, the value of a variety of learning styles becomes even more significant. Logo provides a way of developing those strategies through a special environment.

We invite you to enter the world of Logo! Enjoy and share your experiences and the experiences of others.

WHAT IS A LOGO ENVIRONMENT?

Whether you have one computer in the back of the classroom or access to a collection in a computer lab, the Logo environment remains the same. Students should feel free to explore, question, share information, dabble and grow.

An environment which nurtures the learner pays its dividends in a variety of different ways. There are cognitive nature gains, social and psychological gains. Students who feel free to develop without fear of failure will blossom in this type of environment.

HOW DO WE CREATE A LOGO ENVIRONMENT?

The topics that follow offer specific techniques and suggestions for developing your Logo environment. Feel free to adapt these suggestions into your own unique environment, for each classroom has an existing environment which depends on the "chemistry" between teacher and students.

WHAT IS THE TEACHER'S ROLE IN THE LOGO ENVIRONMENT?

Each teacher who creates a Logo environment in the classroom assumes a dual role. A teacher becomes a learner as well as a facilitator.

As a learner, the teacher should be comfortable with an incomplete knowledge, realizing that that knowledge will be enhanced by a variety of interactions. The teacher, too, ought to feel free to explore, question, and dabble, to share information with students as well as collegues. The teacher needs to be an active participant as a learner, with the computer and with students as teachers.

As a facilitator, the teacher maintains the Logo environment by providing opportunities to explore by encouraging discussion, by providing time to contemplate, reflect, and revise ideas and by responding to ethical issues raised by the new technology.

ARE THE CHILDREN JUST PLAYING?

The new technology brings with it questions and attitudes. "Will computers help students enter the job market?" "Should video games take time from 'real learning'?" "Is Logo a language for learning?" "Are the students just playing?"

It is the newness of the technology, particularly in education, that makes it difficult to answer these questions at this time. However, that will not stop us from searching for the best ways to apply the technology in education. Much thought has been given to these questions and many good articles and books discuss these issues. One of them is included at the end of this chapter. As time passes and one becomes more familiar with the technology, we bring new perspectives and insights to these questions. In reading, re-reading and working with our students we come closer to the answers.

WHAT ARE OUR GOALS?

Although we may not know all the answers to questions about the new technology, we do have specific goals for implementing our Logo program. These goals will help us nurture our students in a Logo environment. They can be used as a yardstick to measure whether "powerful ideas" are fermenting.

- Provide a Logo environment.
- Equip students with learning strategies that transcend content areas and form the basis for analysis of knowledge.
- Help the learner recognize the validity of different learning styles and appreciate and accept their value.
- Develop self-esteem for the learner.
- Ensure equity with regard to age, sex, race, and economic status.
- Encourage sharing of ideas with respect for their source.
- Develop appreciation of the beauty and structure of the language.

USING THIS GUIDE

The Logo Curriculum guide was created in loose-leaf format. This allows the user to modify, rearrange, add, and subtract topics.

We have seen in the past chapters that the new technology (including the Logo language) is dynamic and flexible. It is for this reason that each article or topic deals with one or more "powerful ideas." As a learner in the Logo environment, the teacher will enjoy exploring these ideas. As a facilitator, the teacher can introduce various topics at the level of the students, choosing to emphasize some aspects of the Logo language according to the needs of the individual classroom. In some cases Early (E), Middle Level (M), and Advanced (A) aspects of the same topic are included in the guide.

You and your students may go beyond the limits of this guide. You may find that Topic 6A is appropriate for your first grade class even though it involves some fairly difficult material or that topic 5E is too difficult for your fifth graders. The loose-leaf format allows you to make it a living and growing document.

We encourage you to reach out and share your experiences, ideas, and discoveries. From time to time this guide may be updated with additional materials from you, your students, and many other sources.

This Guide is designed to be used with various implementations of Logo (i.e., TI, Apple). You might have to make some minor changes as you go through this guide. For example, Atari Logo requires brackets [] around the command STOP. The minor changes needed to utilize this Guide can be researched in your specific Logo Reference Guides and Documentation. There are also commercially sold Logo texts that contain appendices of comparison and conversion tables for the various implementations of Logo.

EARLY CHILDHOOD TOPICS

NOTE TO TEACHERS

Since the Logo computer language was designed as an exploratory one, you should allow your students as much freedom to experiment as possible. Remember, the computer does not teach the pupils, rather it is the pupils who teach the computer.

TEACHER INFORMATION

KEYSTROKES:

The following procedures will enable the Logo turtle to move using single keystrokes. Enter them before you present the turtle to the class and save them on a disk or cassette so they can be used each time Logo is loaded. These keystrokes will enable the turtle to move forward (TO F) and back (TO B) and turn 90 degrees left (TO L) and right (TO R).

```
TO F (Apple) or TO FF ( for TI which will not accept F as a procedure)
  FD 20
END

TO B
  BK 20
END

TO R
  RT 90
END

TO L
  LT 90
END
```

The following are additional commands that you may need or wish to teach to your students:

LOGO COMMANDS (Use full words or abbreviations.)

FD	FORWARD	BK	BACK
RT	RIGHT	LT	LEFT
CS	CLEARSCREEN	PU	PENUP
PD	PENDOWN	PE	PENERASE
HT	HIDETURTLE	ST	SHOWTURTLE
SC	SETCOLOR	CB	COLORBACKGROUND

Beyond Keystrokes

The Keystroke procedure that you used to introduce Logo to your students was designed to make Logo commands as simple as possible.

Using this procedure and the lessons provided in this section will probably give most kindergarteners and first graders a well rounded introduction to computing.

With the use of CS, PU, PD, SC and CB, your students have already gone beyond Keystrokes in using regular Logo commands.

Where you go with your students at this point will depend on your assessment of their ability to deal with more complex concepts.

It may be that there are only a few students who appear ready to go further. The following suggestions are made for use with these youngsters:

1. Introduce the commands without the Keystroke program.

   FD (FORWARD)
   BK (BACK)
   RT (RIGHT)
   LT (LEFT)

   . Students may use the abbreviations or the whole word. Bear in mind that the abbreviation will require less searching on the keyboard.

   . Each of these commands must be followed by a space and a number. Have the students experiment to see what results they can achieve.

   . FD and a number enables the turtle to move forward; BK and a number causes it to move backward.

   . With the commands RT and LT and a number the turtle turns either left or right (RT 135 is a right turn of 135 degrees).

   . When the turtle runs out of room, it will "wrap around" and reappear elsewhere on the screen.

2. Although pupils this age are usually too young to grasp the concepts of circumference and degrees, through experimentation they should be able to investigate other geometric shapes such as the triangle and the circle as well as create original shapes not readily available on Keystrokes. (Lessons on procedures for these shapes are elsewhere in this curriculum.)

3. Once students have discovered how to make different shapes, they can combine them to create interesting pictures and designs.

Introduction to Early Childhood Computer Experience

These lessons are designed to introduce young pupils to the fasinating world of computers and the Logo language. They are divided into three areas:

1. Computer Awareness, which seeks to provide a basic understanding of what computers are and the roles they play in the world today.

2. Pre-Computer Activities are designed to create readiness for hands-on experiences through a wide range of visual, auditory, tactile and physical activities.

3. Computer Activities introduce the students to hands-on experiences and explorations in the Logo language.

The Computer System

Performance Objectives - The students will be able to:

. Name the parts of a computer system.
. Describe what each part does.

Background:

. Prepare labels for each part of the computer system:

- Monitor or Television
- Keyboard
- Expansion Unit (if used)
- Computer
- Cassette Recorder or Disk Drive
- Tapes or Diskettes

Development:

1. Show the computer system to the group.

2. Introduce it part by part:

   . Keyboard.

     - "What does this look like? (a typewriter)
     - "This also has keys with letters and numbers on them." (have the pupils read and locate some keys)
     - "We press the keys to tell the computer what to do."

   . Computer.

     - The keyboard is part of the computer. There are also things inside which sort and store the information we give the computer.

- Monitor or Television.
  - This looks like our TV sets at home. Instead of watching television shows, we use it to get information from the computer.
- Cassette Recorder or Disk Drive.
  - This makes a record or tape of information to tell the computer what to do. Computers can't think. We have to tell them exactly what to do.
- Expansion Unit.
  - This gives the computer more memory to do more things.
- Tapes or Diskettes.
  - These are like books. We keep the instructions to the computer on them.

3. Demonstrate how to turn on and start up the computer system.
4. Type a few letters or words to show how the keyboard works and how characters appear on the monitor.

Follow-up:

1. Have the pupils draw a picture of a computer or someone using a computer.
2. Use these pictures and other computer related pictures and worksheets as part of an ongoing computer bulletin board.

The Computer

Performance Objectives - The students will be able to:

- Recognize a computer as a machine.
- Describe some things a computer can do.

Background:

- Collect and/or have the pupils collect pictures of machines, including pictures of computers.

Development:

1. Display pictures of different machines, including several of people working with computers.
2. Discuss what is going on in the pictures, what machines are being used.
3. Ask students to select the pictures that show people working with computers.
4. Ask: "What are the people doing with the computers? What are some things we can do with computers?" (responses will vary based on experiences)
5. Brainstorm with class "What is a computer?" and use responses to establish a definition of a computer.
6. Computers are machines. They can do many things. Ask: "What are some of the things they can do?" "What are some of the things they can't do?" Discuss and use worksheets 1 and 2.
7. Computers and you:
    - "How are you like a computer?"
    - "How are you different from a computer?"
    - "What can a computer do that you can't?"
    - "What can you do that a computer can't?"

Follow-up:

1. Computers are wonderful machines. They can do many things, but not everything! Wouldn't it be wonderful if you had a computerized machine that could do anything you wanted? What would that be? (discuss possibilities) Draw a picture of your "Wonderful Machine" and tell what it can do.
2. This is a good time to start a computer folder for each child to serve as an ongoing record of computer activities.

How Computers Work

Performance Objectives - Students will be able to:

- Describe how an input-output system works.
- Name several input-output systems.

Development:

1. Review the parts of the computer system.

2. Explain that computers work in a special way: as an input/output system.

3. Compare this to how a soda machine operates: You put money in--out comes soda.

4. Play an Input-Ouput game by giving the class something simple to memorize for example, a short poem. Then ask them to recite it back to you.

5. Discuss how the input was what they heard, the brain processed it and stored it, and the output was what they recited.

6. Go to the computer and demonstrate that that is how the computer works.

   - The keyboard is used to send input to the computer.
   - The information is stored in the computer's memory.
   - The output appears on the screen.

7. Demonstrate by typing in a simple program: (this is a BASIC program that will display students' names)

```
10 REM This program prints a list of names
20 REM Substitute names of students
30 ?:? "Lewis ", "Lisa ", "Steve "
40 ?:? "Jason ", "Tom ", "Emily "
50 ?:? "Angel ", "Sarah ", "Sal "
60 END
```

   - Run the program to show the output. Clear the screen and run it again to show how the computer "remembers" the program.

Follow-up:

1. Discuss other input-output systems the pupils are familiar with.

3. Pre-Computer Concepts.

. Note: These lessons are designed to provide students with the readiness and skills needed for computer activities. However, this does not mean that all the pre-computer lessons must be taught before beginning to use the computer itself. You may choose to teach the concepts of directionality and then do related material on the computer. Later, you can return to the pre-computer lessons as needed.

Playing Turtle

Performance Objectives - The student will be able to:

- Physically demonstrate understanding of the directions forward, back, right and left.

Background:

- Make or locate a large circle on the classroom or gym floor.
- Cut out oaktag "turtles" (triangles).

Development:

1. Tell the pupils that a turtle will appear on the screen that can help them draw.

2. Introduce them to the computer's turtle by typing TELL TURTLE and pressing the Enter key.

3. Discuss the turtle that appeared on the screen. Was it what they expected? What does it look like? What shape is it?

4. Explain that the computer's turtle is a shape called a triangle. It may not look like a real turtle, but we can tell it to move forward, back, left and right.

5. Before we use the computer's turtle, we are going to pretend to be turtles ourselves.

6. Use the large circle on the floor:

   - Have a child be the turtle and stand in the middle; have the child move forward and back, following verbal directions.

   - Have several pupils try the following.

     - With a student in the center, explain that the computer's turtle does not move left or right, but turns to the left or to the right.

     - Have the pupils practice turning left and right (90 degrees) and then moving forward and back.

7. Lead the pupils in marching to directions like: "Forward march 1 2 3 4, right turn, march back 1 2 3 4 , turn left, forward march 1 2 3 4..."

8. Discuss what happens when you make two right or two left turns, or four right or left turns.

9. Tape the cut-out turtles on the students' pencils and use grid paper for tracing verbal directions.

Directionality: Movements Forward and Back

Performance Objectives - Students will be able to:

- Follow verbal directions to move forward.
- Follow verbal directions to move back.

Development:

1. Play games to develop understanding of directions forward and back.

   - Giant Steps.

     - "Move 3 giant steps forward."
     - "Move 5 baby steps back."

   - Follow the Leader.

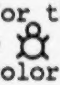
   - After introducing the games, use students as the leaders.

2. Give verbal directions in the classroom to make simple measurements, for example: "How many steps forward to the chalkboard?" and "How many steps back to the sink?."

3. Introduce the concept that forward and back is relative to where you are, depending on the direction in which you are facing. Use worksheet 5 to reinforce this concept.

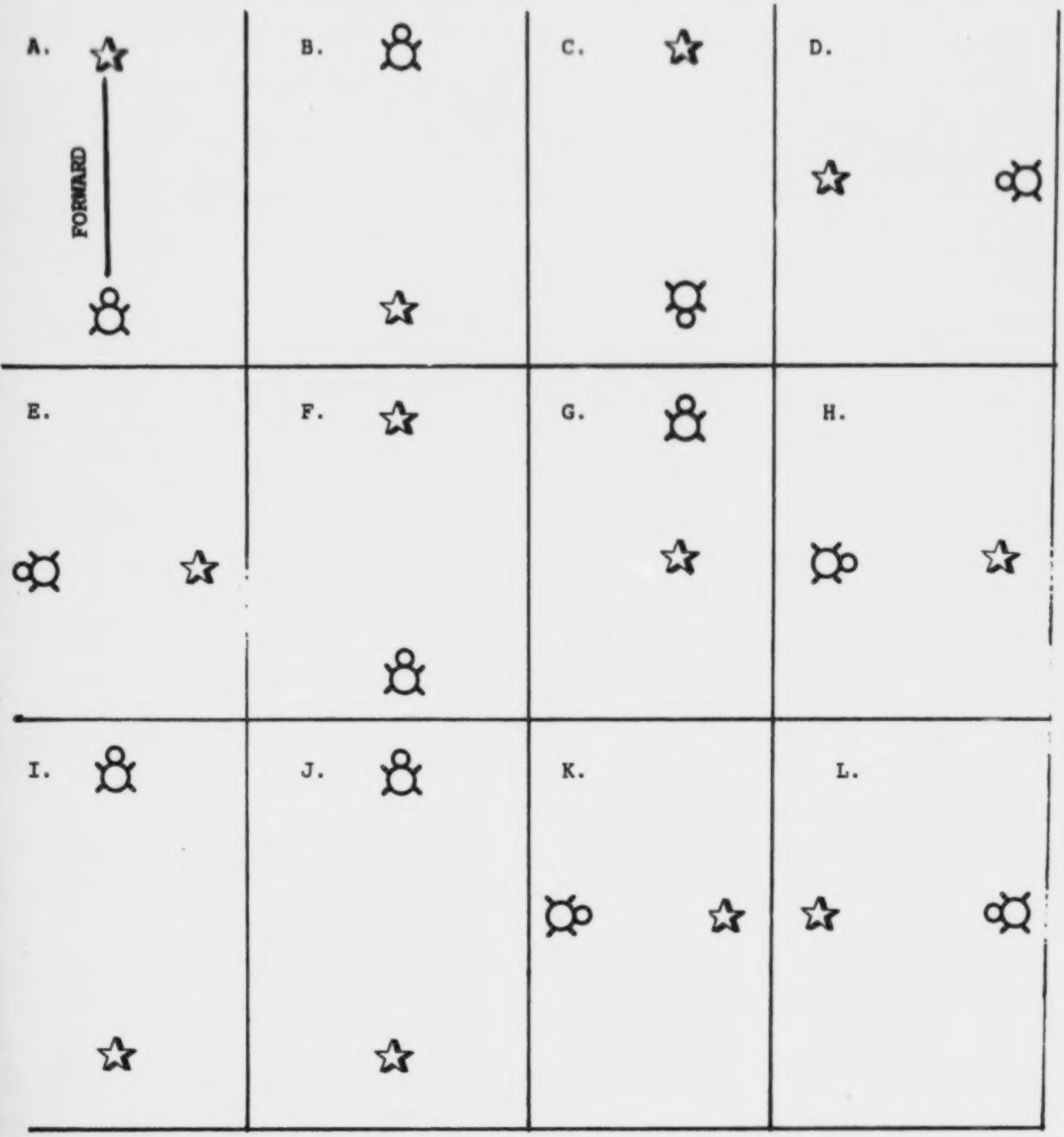
Follow-up:

1. Continue to use a variety of games and exercises to reinforce forward and back.

2. Use worksheet 1 as a paper and crayon exercise for this concept. Have the pupils place a crayon on the box with the turtle . Direct them to go forward or back a given number of boxes and to color in the box on which they stop.

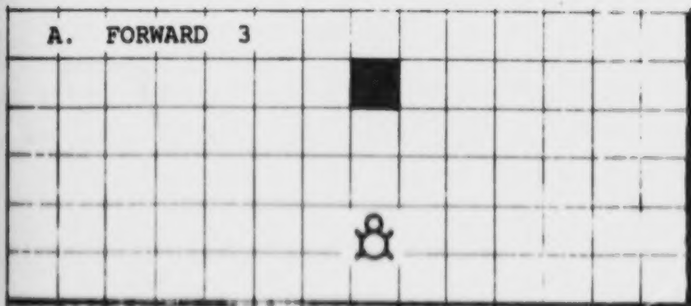
WORKSHEET 1

Directions:
In which direction must each turtle move to reach its star ☆? Some will move FORWARD. Some will move BACK: Draw a line showing the direction each turtle will move. On the line, write the direction: FORWARD or BACK.

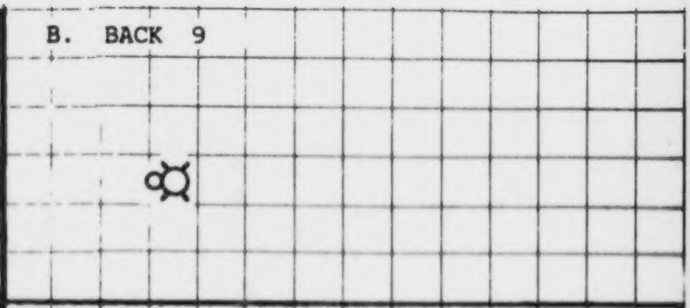


WORKSHEET 2

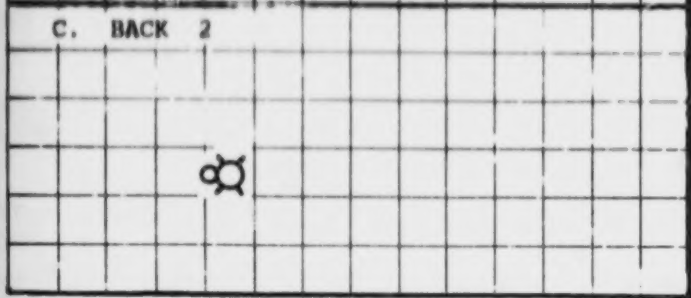
A. FORWARD 3



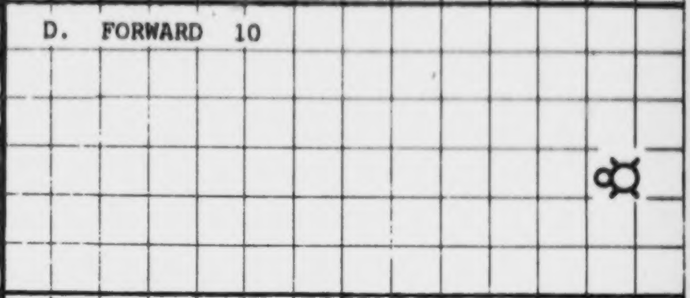
B. BACK 9



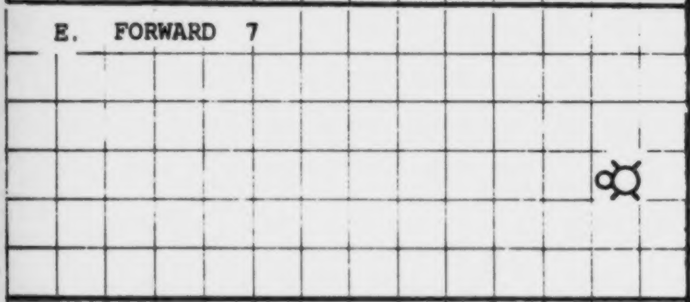
C. BACK 2



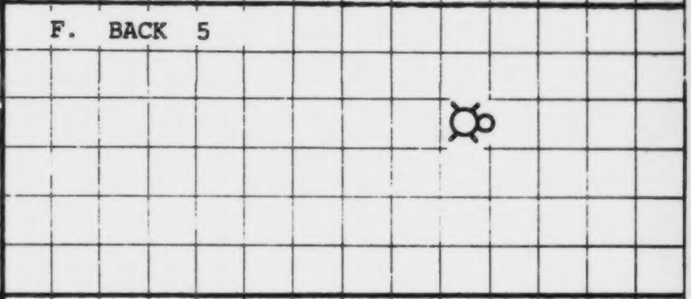
D. FORWARD 10



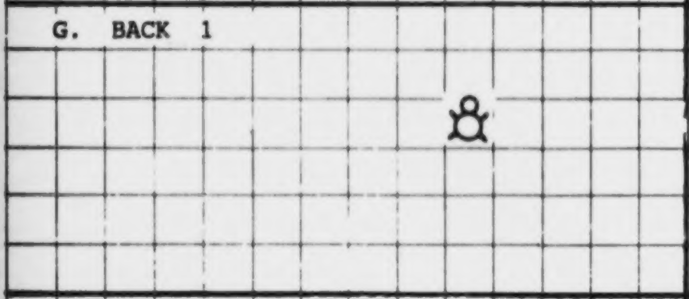
E. FORWARD 7



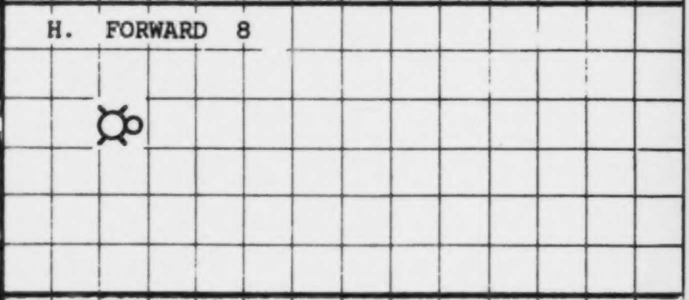
F. BACK 5



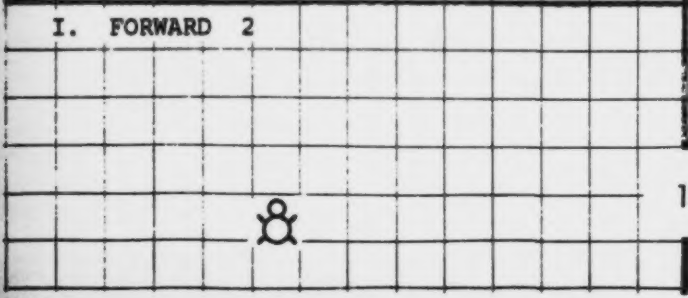
G. BACK 1



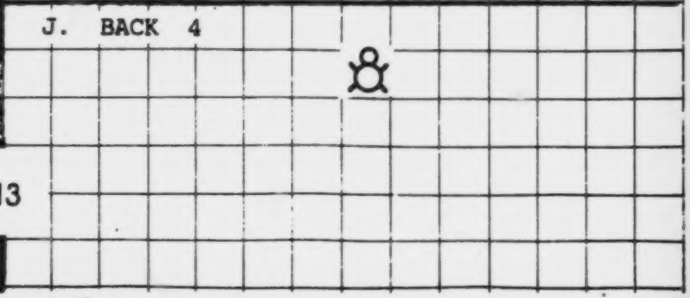
H. FORWARD 8



I. FORWARD 2



J. BACK 4



Directionality: Movements Left and Right

Performance Objectives - Students will be able to:

- Follow verbal directions regarding left and right.
- Demonstrate the ability to follow directions forward, back, left and right.

Development:

1. Teach the concepts of left and right through a variety of activities:

   - Tie a colorful ribbon on each child's right wrist.

   - Give verbal directions using right and left, reminding the pupils that the right hand has the ribbon on it and the left hand does not.

     "Raise your right hand." "Tap your head with your left hand."
     "Move 2 steps to your left." "Point to the person on your right."

   - Have the pupils work with manipulatives on their desks, for example: Move the red block up to the right, move the green block down to the left, place the yellow block on top of the block on the right, etc.

   - Use circle games and dances, for example: Shake the hand of the person your right, bow to the person on your left, skip to the right and stop, slide to the left and stop, etc.

2. Combine with forward and back.

   - Use verbal directions such as: move 3 steps forward, move 2 steps left, move 5 steps back, move 4 steps right.

   - Continue to use games like Follow the Leader.

   - Use mazes and verbal directions (see worksheet 3).

Follow-up:

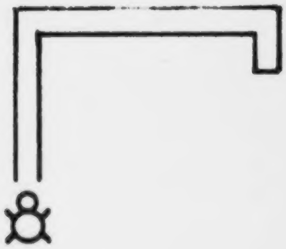
1. The concepts of left and right will require a good deal of repetition and reinforcement through varied activities for many students.

2. Use worksheet 4 and materials from other sources (Kindergarten curriculum, math materials, etc) to reinforce right and left directionality.

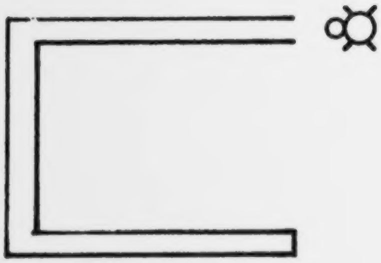
WORKSHEET 3

Directions: Listen to the commands that will teach the turtle to move around the mazes. With your crayon (pencil, finger) trace the turtle's step as I give the commands.

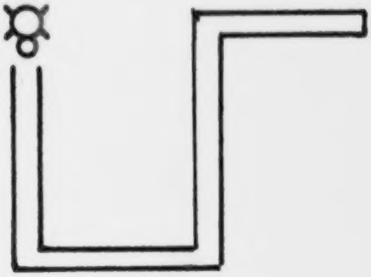
A) FD RT FD RT FD



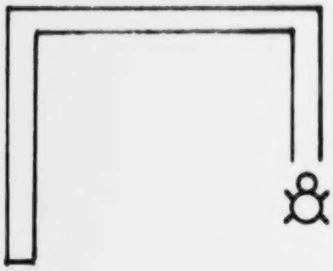
B) FD LT FD LT FD



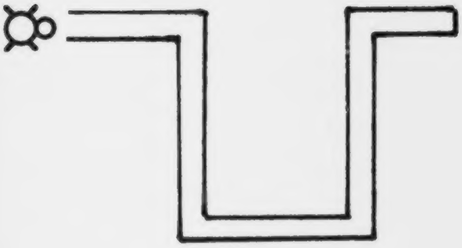
C) FD LT FD LT FD RT FD



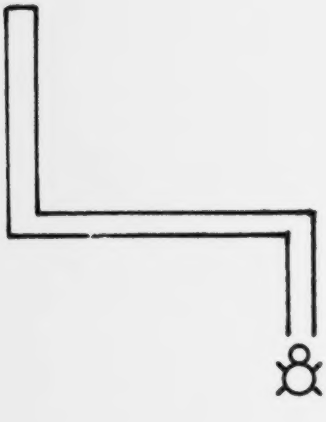
D) FD LT FD LT FD



E) FD RT FD LT FD LT FD RT FD

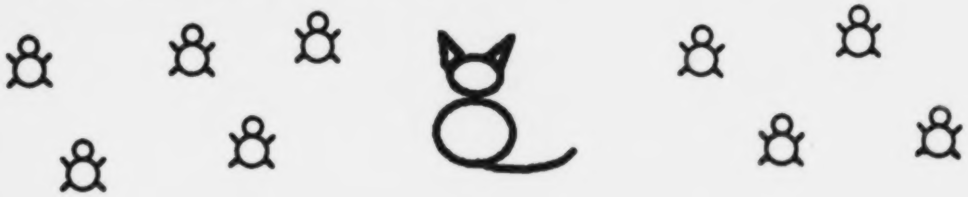


F) FD LT FD RT FD

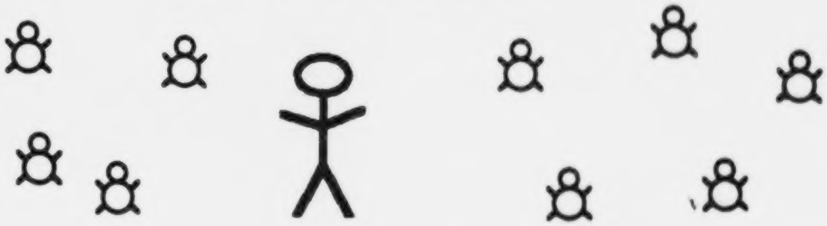


WORKSHEET 4

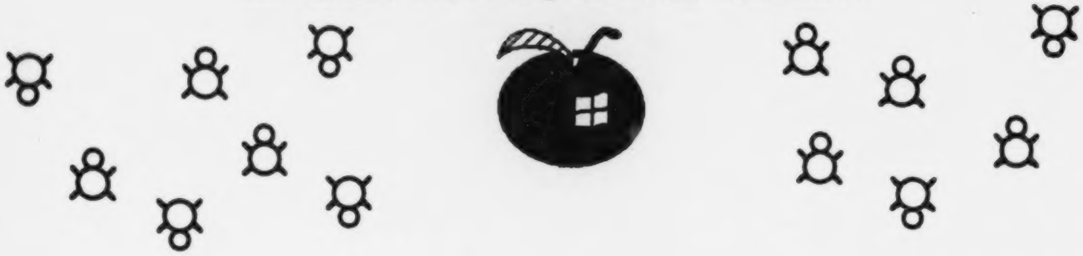
A. Directions: Circle the turtles that must turn RIGHT to reach the cat.
Put x's on the other turtles.



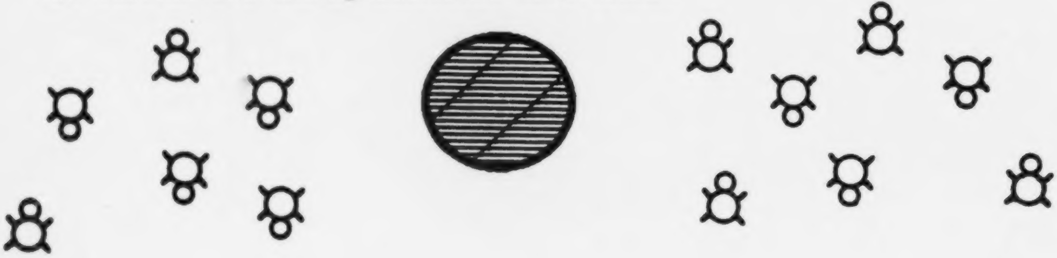
B. Directions: Circle the turtles that must turn LEFT to reach the man.
Put x's on the other turtles.



C. Directions: Circle the turtles that must turn RIGHT to reach the apple.
Put x's on the other turtles. Be careful!
The turtles are facing different directions.



D. Circle the turtles that must turn LEFT to reach the ball. Put x's on the other turtles. Be careful!
The turtles are facing different directions.



Moving the Turtle Forward, Back, Right and Left

Performance Objectives - Students will be able to:

- Locate the F, B, R, L and Enter keys.
- Move the turtle forward, back, right and left on the computer.
- Introduce the CLEARSCREEN command.

Background:

- Load the KEYSTROKE procedure on the computer and proceed to the opening screen for Logo (for TI type TELL TURTLE).
- Place small colored dots on the F, B, R, L and Enter keys.

Development:

1. Ask the pupils what they see on the monitor or screen.
2. Remind them that this is the computer's turtle.
3. Say: "Remember when we played turtle? We moved forward, back, right and left. We can press certain keys to have the turtle do this."
4. Ask the students how the keyboard is different from the last time they saw it. (They should notice the tape on the five keys.)
5. Indicate and name the F key. "Can you think of some words that begin with the letter F?" "F is for FORWARD. Let's see what happens when we press the F key."
6. Have a volunteer press the F key once (twice for TI Logo) and describe what happens. (nothing)
7. Explain that in order for the turtle to follow a command we must do something else.
8. Now locate the Enter key and have the student press it.
9. Discuss what happened. Introduce the CLEARSCREEN command (CS) as the way to erase the screen. Type CLEARSCREEN (CS).
10. Have several pupils try F and Enter. CLEARSCREEN after each student has pressed Enter.
11. Ask what we should do if we wanted to move the turtle BACK. "What letter does the word back begin with?"
12. Have students experiment using B and Enter. CLEARSCREEN after each student has pressed Enter.
13. Have students experiment with moving the turtle both forward and back. CLEARSCREEN after each student has experimented.

(Part 2...Moving Right and Left)

1. Review procedures for moving the turtle forward and back.
2. Remind students of the need to Enter each command.
3. Discuss what we should do if we want the turtle to change direction.
4. Have a volunteer act as the turtle. Have him/her go forward and back. Ask: "How can you change the direction in which you are going?" (Turn right or left.)
5. "Let's do this with the computer."
6. Introduce the R and L keys in the same way that you introduced the F and B keys. CLEARSCREEN after you discussed what happend.
7. Have several pupils experiment with left and right turns. (What happens when they make two lefts or two rights or four lefts or four rights?) Remember to CLEARSCREEN after each student has experimented.
8. Now have the students use all four keystrokes F, B, R and L (and don't forget Enter). Remember to CLEARSCREEN after each student has experimented.

Practice with Forward, Back, Right and Left

Performance Objectives - Students will be able to:

- Use F, B, R, L to create a given design on the screen.
- Use F, B, R, L to reach a given point on the computer's screen.

Background:

- Prepare several transparent sheets with simple designs, one design per sheet.

Development:

1. Review the location and use of the F, B, R, L and Enter keys.

2. Have the students play a game called "Reaching the Star."

   - Place a star near the upper right hand corner of the screen.

   - Ask a student to direct the turtle to reach the star.

   - Discuss the route taken to reach the star.

   - Clear the screen (CS) and have another student try a different route.

   - Move the star to a different location and repeat the procedure with other students.

3. Direction Designs:

   - Place the transparency on the screen.

   - Ask a volunteer to direct the turtle to follow the path and trace the design.

   - Give each student in the group an opportunity to make a "turtle tracing."

Creating Designs Using Forward, Back, Right and Left

Performance Objectives - Students will be able to:

- Move the turtle FORWARD, BACK, RIGHT, LEFT.
- Create original designs using these commands.
- Reinforce and use the CLEARSCREEN command.

Development:

1. Demonstrate how by using the F, B, R, L and Enter keys you can create a design.
2. Review the CLEARSCREEN command (CS) as the way to clear the screen for a new design.
3. Have students work in pairs to create original designs.
4. Have the partner keep a record of the design made. This can be done by writing down the commands as they are made (for example, FFF R FFF R BB L BB L FFF), by drawing a picture of the completed design, or both.
5. Provide opportunities for the pupils to share their creations with their classmates.
   - Discuss how the design was made and what it looks like to the creator and the viewers.
6. Some pupils may wish to try to replicate each others' designs. This is normal and should be encouraged.

Introducing Geometric Shapes

Performance Objectives - Students will be able to:

- Identify simple geometric shapes.

Background:

- Gather materials that are square, rectangular, triangular and round in shape. (blocks, game pieces, pattern blocks, etc.)
- Make a large oaktag cutout of each of the four basic shapes.

Development:

1. Introduce the square:
   - Show a large cutout of a square.
   - Ask: "Does anyone know the name of this shape?"
   - "What can you tell me about this square?" (discuss attributes)
   - Turn it in different directions to show that it looks the same from any side because all four sides are the same size.
   - Ask the pupils to identify objects in the room which are square.
   - Have the pupils play turtle and walk a square.
   - Have the pupils select square objects from a group of different shapes.
   - Use body geometry to have the pupils form a square. Ask: "How many pupils will we need to make a square?" Pick a group and let them experiment with making a square by standing next to each other. Can it be done with more or less pupils?
2. Introduce the rectangle:
   - Use the same procedure as above.
   - Compare the rectangle with the square to see similarities and differences.
3. Introduce the triangle and the circle in the same manner.
4. Give varied experiences with recognizing shapes and locating shapes in familiar objects.

Follow-up:

1. Continue to give practice in shape identification in various activities:

   - Have the pupils recognize shapes when using blocks and other manipulatives.
   - Go on a shape hunt around the classroom.
   - Have the pupils cut out magazine pictures representing different shapes.
   - Find shapes in the letters of the upper case manuscript alphabet.
   - Use colorful paper shapes to create pictures and designs.

Creating Line Designs on the Computer

Performance Objectives - Students will be able to:

- Move the turtle to a specific target on the computer.
- Create a line design on the computer.

Background:

- Gather small colored dots, scotch tape, large colored circles and masking tape.
- Load KEYSTROKE program.

Development:

1. Place two colored circles labelled 1 and 2 on the floor two feet away from each other.
2. Select a student to walk from one circle to the other. Place a strip of masking tape over the student's steps.
3. Have the remainder of the group observe the actions of the selected student.
4. Discuss their observations.
5. Repeat the above procedure several times until the group observes a selected student walking right, left, forward and back to create line designs.
6. Now place 3 (or more) colored, labelled circles on the floor.
7. Select a student to walk from circle 1 to circle 2 to circle 3 (etc.) Place a strip of masking tape over the student's steps.
8. Have the remainder of the group observe the actions of the selected student.
9. Discuss the observations.
10. Repeat the above procedure several times until the group observes a selected student walking right, left, forward and back to create line designs.
11. "Let's try this on the computer."
12. Place a small colored dot labelled 1 on the screen above the turtle.
13. "How can we make the turtle reach the dot."
14. Discuss the procedure. Type B, F, R, L to test hypotheses. CLEARSCREEN.

15. Place the small colored dot in several locations away from the turtle (ie. right, left, back, forward).

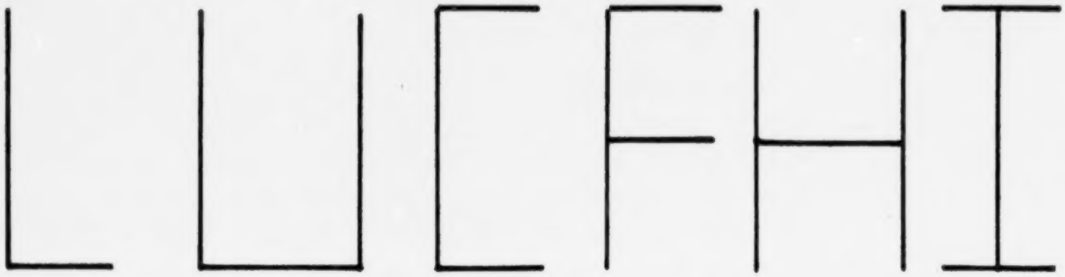
16. "How can we make the turtle reach the dot."

17. Discuss each procedure. Test the hypotheses. CLEARSCREEN after each experiment.

18. Repeat the above using 2 or more colored labelled dots (1,2,3,etc.)

Follow-up:

1. Collectively write directions to create the following upper case letters on the copmuter. Test the predictions on the computer.



Creating Squares on the Computer

Performance Objectives - Students will be able to:

. Create a square on the computer.

Background:

. Pre-computer Concepts lesson on geometric shapes--Squares.

Development:

1. Have the pupils "walk" a square. Discuss what they did. (five steps forward, right turn, five steps forward)
2. "Let's try this on the computer."
3. Select two students to make a square.
4. Have the remainder of the group assist in the process.
5. Clear the screen (CS) and have another pair make another square.
6. Repeat the procedure.
7. If you wish to have several squares on the screen at the same time, use PU and PD commands to relocate the turtle.
   - PU or PENUP can be used to allow the turtle to be moved to a new location on the screen without having to draw a line.
   - PD or PENDOWN is used when you want the turtle to be able to draw again.
   - In this manner, you can have one student make a square, move the turtle to a new location, have another student draw a square and have both squares on the screen.
8. Repeat the procedure until all pupils have had an opportunity to create squares.

Follow-up:

1. Continue activities that reinforce the concept of squares.
2. Compare the different sizes of squares. Use FD n to create different sized squares on the screen.

Creating Rectangles on the Computer

Performance Objectives - Students will be able to:

- Create rectangles on the computer.
- Use the spacebar.
- Use the FD n command.
- Locate the number keys.

Background:

- Load the keystroke program.

Development:

1. Review the similarities and differences of squares andtangles.
2. Review how to "walk a square" by having a volunteer do it. Emphasize the four equal sides.
3. "Who can show me how to walk a rectangle?"
4. While this is being done, discuss the student's movements and allow others to offer suggestions.
5. Review making a square on the computer.
6. Have a student attempt to make a rectangle on the computer. Again, encourage verbalizing the thought process.
7. Introduce the FD n command as a way to change the number of steps forward the turtle can move.
8. Locate F and D keys on keyboard.
9. Have a volunteer press the FD and Enter keys and describe what happens.
10. Explain that in order for the turtle to follow this command we must do 2 other things.
11. Locate the spacebar.
12. Have a volunteer press the FD, spacebar and Enter keys and describe what happens.
13. Explain that in order for the turtle to follow this command we must do something else.
14. Have a student press FD spacebar 5 and Enter keys. Describe what happens.

15. Repeat this procedure using various number keys (ie. 9, 8, 7...)

16. Have a pupil press FD spacebar 1 0 and Enter keys. Describe what happens.

17. Repeat this procedure using various numerals (ie. 20, 30, 40...).

(Part 2...Creating Rectangles On The Computer)

1. Review the FD n command and the locations of the spacebar and number keys.

2. Review the procedure (command and keys) to move the turtle FORWARD.

3. Have student volunteers move the turtle using the FD n command.

4. In the event that students have difficulty in making the transition to rectangles, you can use a transparency with a rectangle for the pupils to trace.

5. Allow each pupil to create at least one original rectangle and to have the opportunity to observe the others at work. CLEARSCREEN after each student creates a rectangle.

6. They should have the chance to see and discuss the variety of rectangles that can be created and the attributes that they possess that make them all rectangles.

Follow-up:

1. Use ongoing activities to reinforce the difference between a square and a rectangle.

Creating Designs with Squares and Rectangles

Performance Objectives - Students will be able to:

- Create original designs on the computer using squares and rectangles.
- Use the PU and PD commands.

Background:

- Review attributes of squares and rectangles.
- Review procedures for making squares and rectangles on the computer.

Development:

1. Introduce the PU and PD commands.

   - Draw a rectangle on the chalkboard, return to the starting point, and keep the chalk there.

   - Then ask, "What should I do if I wanted to draw a square above the rectangle?"

   - Someone will eventually tell you to pick up the chalk and move it to where you want to draw.

   - Follow the directions, saying, "I pick up the chalk when I don't want to draw and I put it down when I want to draw again."

   - "Let's see how we can do this on the computer."

   - Have a student make a rectangle on the computer.

   - "Now, if we want the turtle to move without drawing a line, we have to tell it to pick up the pen the way you told me to pick up the chalk. We tell the turtle PENUP or PU."

   - Have a student type in PU. (Remember to press Enter.) Then have a student move the turtle to where the square is to be drawn.

   - "Can we make a square now?" "If we told the turtle to PENUP, what should we tell it to do so it can draw?" (PU or PENDOWN).

   - Have a student type PD and make a square.

   - Do this several times until the pupils are familiar with the new commands.

2. Working in pairs, have the students make designs that combine squares and rectangles.

3. Have the students use the PU and PD commands to move the turtle. Utilize the whole screen.

4. Students should draw a pictorial record of their designs for their computer folders.

Adding Color to Turtle Designs

Performance Objectives - Students will be able to:

- Use the SC command to add color to their designs.
- Create original designs using lines, squares, rectangles and colors.

Background:

- Prepare a large Logo Color Chart showing color, color name and color number (see attached).

- Note: There are 16 colors available on the computer, you may decide to use fewer colors or only those with lower numbers based on the ability of the students.

Development:

1. "What color does the turtle use to draw?"

2. "We can change the color that the turtle draws with by entering a new command."

3. "When we type SC that means SETCOLOR."

4. "What color would you like the turtle to use?"

5. Show the color chart and have a student find the number of the desired color.

6. "Type SC 6 for red. Now have the turtle draw something red."

7. Continue by giving different students an opportunity to locate the number for different colors. Use SC and the number they located to change the color and experiement with having the turtle draw in that color.

8. Leave the color chart posted near the computer so that the pupils can use it when working at the computer.

Changing Background Colors

Performance Objectives - Students will be able to:

- Use the CB (COLORBACKGROUND) command to change the color of the monitor's screen.

Development:

1. Review how to use SC (SETCOLOR) and a number to change the color the turtle used to draw.
2. Ask: "What color is our monitor (or TV) screen?"
3. "Would you like to learn how to change that color?"
4. Using student-drawn pictures around the room, explain: "When we draw, the part of the picture that is set back is called the background."
5. "When you want to change the background color of the screen, use CB (COLORBACKGROUND) and the number of the color you want.
6. Refer to the color chart. "What would you type in if you wanted to make the background orange?" (CB 9) Have a student type this in.
7. Give students the opportunity to experiment with changing background colors.

Follow-up:

1. Students should have ample time at the computer to combine all the skills they have acquired: using the keystroke programs F, B, R, and L; creating squares and rectangles; using the commands PU, PD, SC and CB.

ELEMENTARY GRADE TOPICS

TOPIC 1

What is a Computer?

Performance Objectives - Students will be able to:

- Define a computer as a tool used for calculating and solving problems.
- List seven specific capabilities of a computer.
- Identify five tasks a computer can perform.
- Recognize five tasks a computer cannot perform.

Vocabulary: compute, computer, calculate, user.

Materials:

- Three common tools familiar to students (shovel, hammer, saw).
- A computer, or a picture of one.

Motivation:

- Have tools set up in front of your. Hold up one of them and ask the children what it is, where they have seen one, what it is used for. Example: "What is a shovel used for? What if we didn't have shovels?"

- Repeat this for each of the remaining tools.

- Establish that these are all tools and that tools make our work easier, get it done faster, and enhance accuracy.

Aim: What is a computer?

Development:

1. Bring out the computer. Once the students identify it, ask such questions as: "What is a computer? Is it a tool like the other objects?"

    - Do not accept any answers. Instead, instruct the students to write their definition of a computer in one or two sentences.

        - Do not allow dictionaries.

        - If they have not yet mastered writing skills, have them verbalize definitions.

2. Have the students read their definitions. Point out the differences among their ideas.

    - Encourage responses like: "We don't know enough about computers; we have not had any experience with computers."

3. Explain that there is a need to learn more about the computer.

4. Compare the computer to a tool. Explain that as the shovel makes it easier to dig a hole, a computer makes solving certain types of problems easier by working quickly and efficiently.

   . Explain that a computer needs to be told what to do by the person using it (user).

5. Re-ask the question: What is a computer? Make a list on the board or on a chart for permanent viewing in the room.

   . Computers Have Special Abilities:

     - They do arithmetic (addition, subtraction, multiplication, division).
     - They save information.
     - They organize information.
     - They find information.
     - They move words.
     - They draw pictures.
     - They play games.

Summary:

1. What is a computer?

Assignments:

1. Have the students write an essay: "I Think the -------- Is the Most Useful Tool Ever Invented." (They do not necessarily have to write about the computer.)

2. Why do you think people are always looking for "the easy way" in doing things? Name some thing you wish were easier for you to do. Think of some ways this might be made easier for you. Explain.

3. Do you think computers have feelings? What similar qualities do humans and computers share? What are some characteristics that are different?

4. Study the phrases below. Then decide which column in the chart each phrase belongs to, and write each under the correct heading. (Discuss the results.)

COMPUTER PROBABLY COULD: COMPUTER PROBABLY COULD NOT:

a) give you the answers to your next Social Studies test
b) add 514,998 + 998,514 and then divide answer by 45
c) babysit for you
d) put the names of all your classmates in alphabetical order
e) lift furniture (because it is a very powerful machine)
f) help you correct a word you misspelled while using it
g) finish ten math examples before you
h) read your mind
i) play tag with you in the park (since it can play games)
j) keep receiving more and more new information without forgetting old information

TOPIC 2

Computers at Home

Performance Objectives - Students will be able to:

- Define specific terms.
- Describe the computer as a communicator in the home.
- Describe the computer as a processor of information.
- List at least five reasons for using a computer at home.

Vocabulary: communication, data, process, processor, online information service, data base

Motivation:

- Role-play a situation where someone is trying to receive information over the phone.
  - A person may be calling a movie theater for the show time of a particular film, the weather bureau, etc.
- Discuss the scene they have just witnessed. Get them to realize and understand that it was more than just a phone conversation. The caller had a purpose, and that was to get information.
- The means of communication used here was the telephone. "Where was the caller most likely calling from?" (his home).
- The computer can also be used as a tool in getting information to us quickly right in the convenience of our own homes.

Aim: How can computers be used in our homes?

Development:

1. Explain that a computer, when linked through a telephone, can communicate with another (even bigger) computer.

2. The home computer receives and processes information from the main computer it is hooked up to right over the phone. This kind of service is known as an online information service. The information contained in the main computer is called data base.

3. List on the board the following examples of information people may be interested in getting from an online information service:

   - stock market reports
   - weather
   - news
   - time

4. Computers can also be used as a household tool in:

   - filing favorite recipes
   - balancing checkbook
   - managing finances
   - playing video games
   - educating (aid in doing homework, writing papers, studying)

Summary:

1. How can the computer be used as a household tool?

Assignments:

1. Find pictures of other means of communication, particularly means used in the home, and prepare a scrapbook or collage.

2. Role-play a salesman trying to persuade a family into buying a home computer.

3. Present the students with this situation: "You have just won a personal home computer in a contest! But, we cannot let you have the computer until you can convince us that it will be put to good use in your home. Write us a letter explaining your reasons for wanting this computer in your home."

4. TASK CARD -- The list below features video games that have not yet hit the market. If you could choose one game to play, which one would you try out? Explain the rules and procedures belonging to the game and describe details using words or pictures.

   - Pillow Fight
   - Zilch!!
   - Buffalo Blockade
   - Gone Bananas!
   - On the Waterfront
   - Ice Cream Invaders
   - Mission Impossible
   - Dragon Demons
   - Volcano Vacuum
   - Wow!!
   - Elephant Kong
   - Ghost Hunt

   - What qualities must a video game have to appeal to you? What are some of your favorites? What makes those your favorites?

5. Interview people to learn why they would like to buy or have bought a home computer. They can then compare their results and perhaps illustrate them on a bar graph.

TOPIC 3

Computers in School

Performance Objectives - Students will be able to:

- Identify four different locations in a school where computers may be utilized to facilitate staff work.
- Explain reasons for using the computer in each of these rooms.
- Describe ways in which the computer can enrich them as students.

Motivation:

- Name "tools" used in the classroom (pens, pencils, board, maps, rulers, books, etc.).
  - If the computer is one of the answers given, ask for an explanation.
  - If it is not given as an answer, bring it to their attention: "Would you consider the computer to be a classroom tool? Could we find ways to use it almost everyday in our class?"

Aim: How can the computer be a helpful tool used in our schools?

Development:

1. Form a chart or list with the students.

IN THE CLASSROOM:

Helps teachers with:

Making general class lists
Testing/evaluating students's work
Preparing lessons
Drill and practice materials
Word processing

Helps students with:

Keeping their own records of progress
Averaging of test scores
Educational activity cards and games in various subject areas
Extra-credit work
Sentence structure
Writing poetry
Forming paragraphs
Solving mathematical problems

2. Where else in the school might a computer be put to use?

3. Continue the chart:

MAIN OFFICE:

Health records
General record information (child's name, address, phone number, parent's name)
Scheduling and setting up new classes
Telephone link-up with Board of Education
Teacher payroll

LIBRARY:

Checking-out borrowed books
Catalog and filing system

CAFETERIA:

Keeping lists of students enrolled in lunch program
File popular lunch menus

Summary:

1. Does a computer have any uses in school?

2. Can a computer help the school run more efficiently?

Assignments:

1. Situation: A school is interested in purchasing computers; two groups of people are arranging a meeting to discuss the issue. One group is in favor of buying the computers; the other is not.

   . Have the students come up with reasons for each group. (Computers will ease the work load; people will lose their jobs; some people are afraid of what they do not understand; etc.)

   . Divide the students into three groups and have two of the groups role-play the situation. Tell them they must try to reach a decision based on what each person speaks about at the meeting. The students who are not involved in the meeting could be the judges.

2. Not every school has a computer. Do you feel lucky and privileged to be able to have at least one in your school? Explain why.

TOPIC 4
Careers with Computers

Performance Objectives - Students will be able to:

- Describe careers involving computers.
- Understand and become aware of the job opportunities available to them involving the computer.

Materials:

- At least ten classified ads featuring computer-related jobs, posted on oaktag.

- Highlight the skills required with a yellow marker.

Motivation:

- Discuss careers.

  - What choices do you have?
  - How do you decide on a career?
  - Do you have to make a decision right now?
  - Should you start thinking about this now, or do you think you should wait?
  - What careers involve the use of computers?
  - Do you think that job will still be available ten years from now?

Aim: How are computers used in careers?

Development:

1. Show the ads. Read one or two of them, emphasizing the computer skills necessary for the job. Discuss each one. You might come across a few special terms like these: programmer, word processor, analyst, keypunch operator, technician. Use them as vocabulary words.

2. Have the students read aloud individual ads.

   - What can you conclude about the relationship between the computer field and job opportunities. Do the ads open doors in terms of future job opportunities?

3. Ask how computers might be utilized in the careers they are interested in.

Summary:

1. Computer related jobs are numerous.

2. The use of computers has created new jobs for people.

3. Computers can be utilized in many careers.

Assignments:

1. Research project: The Effect Of Computers On My Career Choice

2. Research Project: Computers And A Career Around The World

   . How do people from other countries use computers to help them in their job?(i.e., secretary, teacher, doctor, lawyer, police, etc.) Compare their uses with ours.

3. Interview people in various businesses and find out how their jobs have been affected by the computer.

4. Role-play a person applying for a computer-related job. The interviewer should focus mainly on the applicant's interest in computers and experience.

5. Have students illustrate and label scenes of themselves seated at a computer in their future careers for bulletin boards.

TOPIC 5-E

Computers of the Past - Who Developed the First "Computers?"

Performance Objectives - Students will be able to:

- Explain reasons for needing better methods of calculating in the past.
- Identify, describe, and compare early devices used for calculating.
- Identify famous scientists who developed these devices.
- Recognize the relationships between these inventions and modern calculating machines.

Vocabulary: abacus, mechanical, gears, loom, punch card, census, tabulating machine

Motivation:

- Count the number of girls and boys in the class separately, and then to calculate the total number of students in the class as a whole.

- What could be used to count accurately? (fingers and toes, paper and pencil, chalk and board, etc.)

- "Suppose next we wanted to calculate the total number of boys and girls in the entire school, and after that, the number of boys and girls in all of New York City, and then all the students in New York City. Would we use the same methods as we named before, or would we use some different devices?"

- Why are adding machines, calculators, and computers more appropriate in certain cases?

  - The easier the calculation, the simpler the device. Complicated computing problems require more intricate computing machines.

Aim: Who developed the first computing devices?

Development:

1. Explain that the very first computing device did not have to be created--it was the "fingers."

2. Scientists realized that as mathematical problems became more complicated, there was more of a need for special machines to do calculations.

   - Until such machines were invented, it took people a very long time to perform mathematical operations.

3. Pretend the following excerpts are from diaries written by five famous scientists who are important to the history of the computer. Have the students read each one aloud and then discuss it.

. Questions for discussion can be found following the diary entries.

. If possible, have drawings of machines mentioned in the diary entries.

Date: 1692

Dear Diary,

The Chinese were real clever when they invented the abacus, but that was thousands of years ago. Sliding beads on a wire has been a great idea; however, it does not help us to compute. I've been thinking of developing an adding machine that would add and subtract numbers through the use of wheels and gears. I know I'm only 19 years old, but what do you think, Diary? Do you really think my idea will work?

Until tomorrow,

Blaise Pascal of France

Date: 1600's

Dear Diary,

If Pascal never invented the first mechanical calculating machine, I wonder if I ever would've created a similar machine that could multiply and divide, as well as add and subtract. Pascal's ideas sure helped me a lot with my own!

Until next time,

Gottfried Leibniz of Germany

Date: 1801

Dear Diary,

You must congratulate me! I've just invented a loom that weaves cloth using punch cards. I know, Diary, this is all new to you, so I'll explain. The punch cards have holes in them allowing threads to go through where they belong in a particular weaving pattern. Thus, I've improved on the loom and the entire weaving industry!

Until tomorrow,

Joseph-Marie Jacquard of France

Date: 1834

Dear Diary,

Have I got news for you! With my assistant, Miss Ada Byron, I have invented what I call a "Difference Engine." It accurately performs calculations of big numbers. Miss Byron helped me develop the special instruction cards that program the machine. My machine is the first to be able to process and store information!

Until next time,

Charles Babbage of England

Date: 1890

Dear Diary,

Well, it's 1890, and could you believe we just got back the results of the 1880's U.S. Census. That was 10 years ago. Why should we care about how many people were living in the United States then, when this is 1890? And who's going to care about the results of the 1890 U.S. Census 10 years from now, if it's going to take that long? There's got to be a better way to count up the figures, Diary, and I believe I've come up with it. Here's my idea:

It's a tabulating machine that would read information (data) recorded on a punch card. Each hole in the punch card would represent a piece of data about someone participating in the U.S. census.

Well, what do you think?

Until tomorrow,

Dr. Herman Hollerith

The Diary apparently thought the it was a great idea, because with Dr. Hollerith's tabulating machine, the 1890 Census was calculated in just 3 years, instead of 10.

Applications:

1. What was Pascal referring to when he mentioned "beads on a wire?"

2. Did Pascal actually go ahead with his idea? How do you know?

3. In what ways do you think Jacquard's loom changed the weaving industry?

4. Why do you suppose Babbage named his machine the "Difference Engine?"

5. What do you think of Dr. Hollerith's "better way?" What problems in our present society need to be solved in a better way?

6. How are these scientists' inventions related to each other? Perhaps they are not at all. What do you think?

7. List specific words to describe these men. Whom do you admire most? Why?

Summary:

1. In the past, the realization for simpler and quicker methods of calculating was the basis of computer development.

Assignments:

1. Imagine you are given the opportunity to interview the following people who made significant contributions to the development of the computer.

    - Blaise Pascal
    - Joseph Jacquard
    - Charles Babbage
    - Gottfried Leibniz
    - Ada Byron
    - Dr. Herman Hollerith

2. Prepare two questions to ask each person. Focus on them and their ideas. What would you want to learn about? What do you want to know?

3. Research Napier's Bones. Define, describe and list the uses.

What Were the First "Real" Computers Like?

Performance Objectives - Students will be able to:

- Describe the present day computer as a descendant from a large family.
- Describe the direction taken in computer development.
- Recognize the progress shown through the historical development of computers.

Vocabulary: Descendant, digital computer, electro-mechanical, electronic, vacuum tube, transistor, integrated circuit, integrated circuit chip.

Motivation:

- Illustrate family trees, including grandparents, parents, and siblings. Outline one on the board for the students to follow as a guide.

- Explain that people are descended from their parents, who descended from their parents, and so on. This makes up their family.

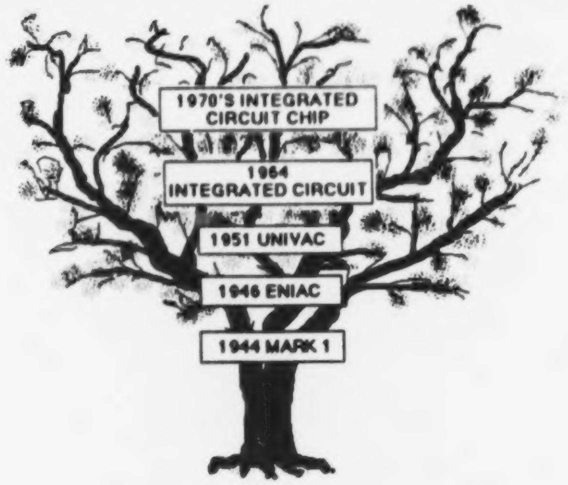
- Computers also descended from "parent" computers.

Aim: What were the first "real" computers like?

Development:

1. As time went on, there was a greater need for machines that could do more than just calculate. People needed a machine that was capable of performing conditional operations, of being easily programmed, and of having long-term memory. What they needed was a computer.

2. Draw a computer family tree on the board similar to the one pictured below. Explanations for each member of family are found below it.



1944 MARK I - First electro-mechanical computer (has electrical and mechanical parts). Huge machine, about the size of a gym.

1946 ENIAC - First completely electronic computer. Used in World War II.

1951 UNIVAC - First commercial computer. Conducted electricity though vacuum tubes.(switches current on and off). Late 1950's Transistor's - replaced vacuum tubes; conducted electricity faster; rarely needed to be replaced.

1964 Integrated Circuit - replaced transistor; smaller, took up less space.

1970'S Integrated Circuit Chip - Thousands of circuits on a chip so tiny the circuits are microscopic. Capable of 10,000,000 calculations per second! Illustrate a single tiny chip to give students a better idea.

3. After the tree is done, it is important to explain that computers were getting much smaller as they were "being born."

4. Also explain that these computers were called digital computers meaning they changed information into numbers to be stored and processed.

Summary:

1. Modern computers are the "students" computers of the past.

Assignments:

1. Activity flash cards where students check themselves. If the puzzle pieces match up, then they are correct.

2. People by nature are constantly trying to better themselves. They have evidenced this from prehistoric to present times.

   . An example of this need would be from the caveman's discovery of fire, to the torch, to the oil lantern, to the electric light bulb, or lamp.

3. How is the computer an example of the progress people have made?

4. Do you think humans can continue to advance themselves where computers are concerned? Explain.

TOPIC 6-A

Modern Computers

Performance Objectives - Students will be able to:

- . Identify the important characteristics of each generation of computers.
- . Understand how the computer has changed in size, cost and efficiency.

Vocabulary: digital computer, generation, ENIAC, vacuum tube, transistor, UNIVAC, integrated circuits, chip, microchip.

Motivation:

- . Write the title "Four Generations of Modern Computers" on the chalkboard. Tell students they will be reading an article with that title. Ask them to guess what the article will be about. List all guesses on the chalkboard.

- . Distribute the article.

  - The article appears in the Board of Education publication Computer Literacy Elementary Grades - Experimental, pp. 165-170 and in Computer Literacy - Intermediate and Secondary Grades - Experimental between pages 12 and 13. A print copy is attached.

- . Have the students skim the article to see which of their guesses were correct and which were not. Erase any incorrect guesses and add new items to the list.

- . Pass around actual computer parts of each generation and explain:

  - First Generation - Vacuum tubes
  - Second Generation - Transistors
  - Third Generation - Integrated circuits
  - Fourth Generation - Use of chips and microchips

Aim: How have computers changed since the first computer was built in 1944?

Development:

1. Read the article.

2. How do today's computers differ from earlier computers?

3. What might future computers be like?

4. What advances might take place in computers for them to become as common in people's homes as television sets are today?

5. Why do you think the ability to store a program was a very important advance in computer technology?

Applications:

1. The price of computers has fallen about 50 per cent every two years. A large microcomputer costs approximately $5,000 today. If the trend continues, how much will a similar computer cost in two years? In four years?

2. Some home computers cost less than television sets. Do you believe that everyone who can afford to buy a computer will do so? Explain your answer.

3. Research and report on fifth generation computers.

4. Develop a time line which illustrates the chronological development of the four generations of computers.

Summary:

1. How have computers changed since 1944?

2. Predict what new changes in computers we can expect in the near future.

Assignments:

1. Complete the activities listed below the article "Four Generations of Modern Computers."

2. Pretend you are a salesperson. Design an advertisement for a computer of each generation. Include a description of the computer and explain how it will help people. Include the advantages it offers over an earlier generation of computer, where applicable.

TOPIC 7
The Computer as a Problem-Solving Tool

Performance Objectives - Students will be able to:

- Recognize the speed of a computer to do math problem-solving.
- Understand that a computer is designed with math logic inside.
- Recognize the computer symbols for addition (+), subtraction (-), multiplication (*), and division (/).

Vocabulary: logical, calculation, memory, keyboard.

Motivation:

- Have a student do a math problem at the board. Have another student record the time. (768 x 397).

Aim: The computer as a problem-solving tool and how it is able to do mathematics.

Development:

1. Record the time it took for the student to do the math problem.
2. Explain and demonstrate the symbols for addition, subtraction, multiplication and division on the computer keyboard.
3. Demonstrate the math problem, 768 x 397 on the computer. Have a student record the time. Compare the difference.
   - Do other problems under other operations and record the times.
4. Why is the computer faster or not faster?
5. How does the computer know how to do math?

Applications:

1. Have students work on math problems at seats. Time the speed of the completion of one or more of the students. Check the accuracy of each answer.
2. Use a calculator to do the same math problems. Time the speed and compare.
3. Use a computer to do the same math problems. Time the speed and compare.

Summary:

1. Which is faster, a computer or a calculator?
2. How does the computer and calculator know how to do math?
3. What are the symbols for addition, subtraction, multiplication and division on the computer keyboard?
4. What human factor(s) could slow down a calculator and/or computer?

Assignments:

1. Write a composition on other uses a computer can have to solve problems.
2. Where might a computer be used to solve problems?

TOPIC 8 - A

Computers Ethics - What's Right?

Performance Objectives - Students will be able to:

- State the correct uses of computers in our society.
- Identify incorrect uses of computers in our society.
- Recognize the need for special security on certain computer systems.
- Recognize the dangers computers can place on the world.

Vocabulary: hacker, tourists, crashers, data banks, ethics

Motivation:

- Discuss the movie "War Games."
  - Did the students have the right to break into the school computer and change their grades?
  - Did the students have the right to break into other computers to steal information?
  - What are the consequences?

Aim: What is computer ethics?

Development:

1. What does ethics mean?
   - The discipline dealing with what is good and bad and with moral duty and obligation; the principles of conduct governing an individual or group.
2. How are computers used to help us?
   - Solve problems.
   - Keep track of information.
   - Play games.
3. What are data banks?
   - Data banks, also called data bases, are large computer systems that store large amounts of information. Government agencies, hospitals, schools, businesses, credit bureaus, banks, and social agencies maintain data banks.
4. Who owns the data in a data bank--the person who collected it or the person whom the data describes?
5. What is crime? Stealing, murder, etc.?

6. What is computer crime? What are some examples?

   - The illegal copying of software which belongs to a programmer, publisher or computer manufacturer.
   - The spying on long distance telephone calls that are sent via satellite.
   - The stealing of business secrets, research data, and marketing plans by intercepting messages sent from one computer to another.
   - The stealing or changing of information in data banks.
   - Money can be taken from someone else's bank account.

7. Computer criminals:

   - Recently, a hacker was defined as a person who breaks into other people's computers; traditionally, it meant someone who devoted time and energy solving problems at the computer.
   - Tourists are people who penetrate computer systems just to look without disturbing the data.
   - Crashers erase or change files, thereby disrupting organizations and potentially harming the individuals whose records are changed.

8. Invasion of privacy:

   - Information about how you spend your money, what magazines you read, what medicines you take, etc. can be found in data banks and sold to interested parties or accessed by the government.

9. The Privacy Act of 1974, an amendment to the Freedom of Information Act, and the Freedom of Information Act itself are pieces of legislation which can help to protect people from some misuse of computer information.

Applications:

1. Have students list ways in which computer crime can affect their lives.
2. Have students list ways in which the misuse of computer information can affect their lives.

Summary:

1. How does what's right in everyday life compare to what's right with computer use?
2. How are our lives affected by unethical computer practices?

Assignments:

1. Have students list ways to stop computer crime.
2. What new laws are necessary to prevent computer crime?
3. What should the punishment be for committing a computer crime?

TOPIC 9-A

Advantages to Business and Government

Performance Objectives - Students will be able to:

- Explain how government and business can benefit from computer use.
- Explain the types of tasks computers are used for.
- Suggest possible future uses of computers.
- Explain the tasks computers perform.

Vocabulary: UPC universal code, retrieval, data processing, process control, simulation, productivity, espionage

Motivation:

- List some electronic devices that are part of your life.
  - digital watch
  - calculator
  - television with electronic tuning
  - computer/video games
  - stereo radios
  - TouchTone phone
  - cash register
  - microwave oven
  - automatic teller machines
  - VCR's

- How are these things related to computers?

- How have electronic devices improved our lives?
  - speed of obtaining results
  - ability to accomplish more work
  - given pleasure
  - improved efficiency
  - improved accuracy

Aim: What are the advantages of computers to business and government?

Development:

1. What are the tasks that computers perform?

   - Retrieval -- getting information back that is already input for example, checking a bank account balance.

   - Processing -- manipulating data for example, adding numbers together.

   - Process control -- regulating certain situations such as when a thermostat checks the temperature and turns on heat or air conditioning.

   - Simulation -- imitating a real life situation on the computer so as to learn something. Examples are: simulating different flight and emergency plans; simulating the effects of the use of fertilizers, pesticides, irrigation and planting time on crops.

2. What physical improvements have there been?

   - Physical size is down.

   - Speed and memory are up, and their cost is down.

   - Scientists are now even using computers to design new computers which are smaller and faster.

3. How are computers used in the business world and why?

   - Payroll
   - Employee records
   - Marketing strategy
   - Word processing
   - Factory robots
   - Airline bookings
   - Inventory
   - Research
   - Electronic mail
   - Message systems
   - Banking
   - Real estate inquiries

4. Examples of computer use in the business world:

   - Supermarkets -- UPC code, inventory, accounting
   - Factories -- robots, process control
   - Banks -- savings and checking accounts, loan payments, credit history
   - Utilities -- calculating usage and cost, printing bills, monitoring payments, projecting needs and peak periods.

5. How does the government use computers?

   - Internal Revenue Service -- collecting, calculating and refunding taxes
   - Defense -- guiding missiles, simulating war plans, encrypting and decrypting messages, maintaining personnel records
   - NASA -- training astronauts in space flight simulators, collecting and analyzing data from missions
   - Local governments -- car registration and license, traffic control
   - Criminal investigation -- maintaining criminal records, voice prints and fingerprints, and inventory of stolen property
   - Other uses -- distribution of social security and Medicare payments, maintaining census records

Applications:

1. Explain the advantages of using computers in the following:

   - Secretary using a word processing machine.
   - Investigators at the FBI.
   - Using electronic mail in the office.
   - An employee in a fast food restaurant taking a large order.
   - An owner of a gas station.
   - A shopper in a department store waiting on a line to pay.
   - Making a savings account deposit.
   - The space program.
   - Weather forecasting.

2. Describe the type of activity the computer is used for in each of the above (retrieval processing, process control, simulation).

Summary:

1. What are the benefits of using computers in business and the government?

2. Who benefits from them?

3. What are the capabilities that make computers good for these uses?

Assignments:

1. How do businesses improve productivity by using computers?

2. How does government use computers to run a more efficient operation?

3. Are people losing jobs because of computers?

4. Make a collage, scrapbook or chart of employment want ads for computer-related jobs.

TOPIC 10-A

Advantages to Organized Sports and the Creative Arts

Performance Objectives - Students will be able to:

- Explain how organized sports and creative arts can benefit from computer use.
- Explain the tasks computers perform.
- Understand word processing.

Vocabulary: creative, word processing, statistics, manipulate.

Background: Newspapers and magazines containing articles on the use of computers in sports and the arts.

Motivation:

- How have electronic devices (calculators, digital watches, etc.) improved our lives?

  - speed of obtaining results
  - improved accuracy
  - ability to accomplish more work
  - improved efficiency

Aim: What are the advantages of computers to organized sports and creative arts?

Development:

1. Review tasks that computers perform.

   - Retrieval
   - Process control
   - Processing
   - Simulation

2. How can computers be used in organized sports?

   - keeping records and statistics
   - keeping inventory of equipment
   - tabulating scores in the Olympics
   - scheduling games
   - paying athletes
   - recording time in races

3. What are creative art areas?

   - Dance
   - Art
   - Theater
   - Movies
   - Music
   - Creative writing

4. How can computers be used in creative arts?

   - Creative writing

     - For writing stories or scripts, word processing and outlining programs make it easier to type, revise, store, retrieve and print text.

     - For record keeping, a database management program can keep track of story ideas, characters, etc..

. Art

- For animation, drawing and sketching, the computer offers a new media with which to create. Painting programs can be used to sketch preliminary studies. Repetitive designs, such as textile designs, are easily drawn.

. Music

- For playing music, MIDI interfaces allow musical instruments to be used as input devices. The computer itself can be used as an instrument if it comes equipped with a speech synthesizer. Musicians can compose a tune and then have the computer alter the notes, rythm and speed.

. Theater

- Stage lighting can be controlled by a computer. Each change in lighting is put into the computer's memory and as the action of the play changes the computer modifies the lights accordingly. The same lighting pattern is used at each performance.

. Movies

- Special effects are often easier and cheaper to create with computer programs than to build sets and props. In addition to superimposing computer images, computers can be used to control robots and vehicles, as well as cameras mounted on mechanical arms.

. Dance

- Choreographers can simulate their dance patterns to visualize and alter the dance steps before introducing them to dancers.

Applications:

1. Break up into groups and find other uses of computers in organized sports and creative arts by looking at the newspapers and magazines.

2. Explain the advantages of using computers in the following situations:

   . Sports commentator broadcasting a Mets game.
   . The Giants and Jets sharing the same football stadium.
   . A boxer preparing for a championship match.
   . The Yankees preparing for a road trip.
   . Football teams drafting college players.
   . A theatrical lighting designer working on three shows at the same time.
   . A writer doing research for a novel set during the Civil War.
   . A dramatist working on his newest play.
   . A commercial being made for TV.

3. Describe the type of activity the computer is used for in each of the above (retrieval, processing, process control, simulation.)

Summary:

1. How can a computer be used in organized sports and creative arts?

2. What are the benefits of these uses?

Assignments:

1. Write an essay. Discuss how computers would be used in your school for creative arts. Consider the following: Would a certain type of computer serve the needs of all the creative artists? For example, would a computer with superior graphics be used for word processing and database management?

2. Investigate the possibility of the local Little League purchasing a computer. What could it be used for?

TOPIC 11 - A

Advantages in Education, Medicine and Law

Performance Objectives - Students will be able to:

- Explain how computers are used in education, law, and medicine.
- Understand the terminology of CAI and CMI.
- Suggest possible future uses of computers in education.
- Recognize different computer languages.

Vocabulary - New: CAI (Computer Assisted Insruction), CMI (Computer Management Instruction), Logo, BASIC

Motivation:

- How are your New York City reading and math tests marked?

- Do the police use computers? For what reasons?

  - keeping track of criminals
  - solving crimes
  - payroll
  - inventory of weapons

Aim: How are computers used in the fields of education, law, and medicine?

Development:

1. Review - what are the tasks that computers perform?

   - Retrieval - getting information back that is already input: checking your balance in a bank account.

   - Processing - taking data and doing something with it: adding numbers together.

   - Process control - regulating certain situations: a thermostat checks the temperature and turns on heat or air conditioning.

   - Simulation - imitating a real life situation on the computer in order to find the best solution: engineers in the space program simulate different courses for the space ship in order to find the best one.

2. How are computers used in the field of law?

- Legal research.
- Records.
- Law enforcement - ex. radar detection for speeding, exchange of criminal photos by computer printouts.
- Investigation: voiceprints, lie detector, fingerprint records.
- Correspondence - word processing.
- The advantages are that computers make more accurate information available more quickly.

3. How are computers used in the field of medicine?

- Specialists can treat patients in faraway places.
- Hospitals - intensive care units monitor body functions, heart rate, blood pressure, temperature, blood analysis, patient medical history.
- Medical diagnosis - ex. CAT scan.
- Preventive medicine.
- Research - to keep records and look for new cures for diseases.
- Medical records.
- Billing and inventory.
- Correspondence - word processing.
- The advantages are that computers make the job more accurate and efficient.

4. How are computers used in the education field?

- Learning from Computers - CAI - Computer Assisted Instruction
  - Drill and Practice, Tutorial

- Learning with Computers:

  - Simulations - enhancing life experiences
  - Games - developing strategies

- Learning about Computers:

  - Computer literacy and awareness
  - Programming languages
  - Word processing

- Learning about Thinking with Computers:

  - Logo - computer language that goes from simple to more complex thinking
  - BASIC - Beginners All Purpose Symbolic Instruction Code

. Managing Learning with Computers:

- CMI - Computer Management Instruction
- Attendance
- Test Generators
- Budget
- Word Processing
- Scheduling of classes
- Communication network between schools
- Diagnosis and prescription for learning
- Inventory of student records

Applications:

1. Explain the advantage of using computers in the following situations:

   . Your school's attendance officer.
   . Your teacher has to mark 123 exams.
   . A student practicing math examples.
   . A hospital patient who needs to have his breath analyzed for proper amounts of oxygen.
   . A lawyer is preparing a case in defense of his client.
   . A patient describes a series of unusual symptoms to his doctor.
   . A doctor must decide which of several available medicines to prescribe for his patient.
   . The Heat Compaint Bureau gets a call from a tenant in an apartment house.
   . Tax laws are changed every year.

2. Describe the type of activity the computer is used for in each of the above (retrieval, processing, process control, simulation.)

Summary:

1. What are advantages of using computers in the medical field?

2. What are advantages of using computers in the law field?

3. What are advantages of using computers in education?

Assignments:

1. Write an editorial discussing the need for future uses of computers in our school. Support your stand by describing what benefits students, teachers and administrators would gain.

2. Interview an attorney or physician to find out how he or she is using computers, and his or her plans for using computers in the future.

TOPIC 12-A

Computer Disadvantages - Privacy and Criminality

Performance Objectives - Students will be able to:

- Define a data bank (data base).
- Outline the process in which a computer acesses a data base.
- Explain how data bases can possibly be used to violate privacy.
- Explain other crimes associated with unlawfully accessing a data base.

Vocabulary: data base, data bank, access, hacker, crusher, tourist, modem, criminal, civil, mainframe

Motivation:

- Narrate the following story to the class:

  Twenty-six students from the California Institute of Technology took the MacDonald's Corporation at their word and entered a contest as often as they wanted to. All a contestant had to do was submit an entry blank or a 3x5 piece of paper with his or her name printed on it. After programming the school's computer, the students printed and submitted over a million entries. Since the entries represented 33% of those submitted, the students won a large share of the prizes.

- Was it fair for the students to have done this?

- Even if they didn't break any of the rules, should they receive the prizes?

- What does this kind of incident tell us about what can be done with computer technology?

Development:

1. What is a data base and how is it used by a computer?

   - A data bank or data base is a systematic electronic storage of information that can be accessed by a computer.

   - Data is information--numbers, dates, heights, addresses, bank balances, etc.--it's all data.

   - From the moment of birth, an individual begins to generate data.

     - For example, your name and birthdate are most likely stored in the data bank used by the hospital's computor. If the hospital wished to find out your date of birth, they could simply use their computer to search through the data base and then print out your date of birth or any other data they have on you.

78

. The same data may appear on many different data bases.

- For example, all students registered in the NYC public school system have their names, addresses and identification numbers on a data base. If you have a bank account or a social security number, you are part of other data bases. As you get older and your life becomes more complicated with jobs, taxes and credit cards, the number of data bases that your name appears on will increase tremendously.

2. How can data bases be misused?

. Large organizations, such as corporations, hospitals and schools, use mainframes, large computers that handle millions of pieces of data.

. These mainframes maintain data bases which are regularly accessed by employees who may be working quite a distance from the mainframe. The data is transmitted and received over the telephone lines.

. This procedure permits authorized users to gain access to the database for analysis or update. Because it was designed to be easy for the authorized user, unauthorized users also find it easy to access the same databases.

3. How do unauthorized users gain access to data bases?

. The problem of unauthorized users gaining access to databases has grown tremendously as the number of microcomputers has grown.

. A microcomputer equipped with a modem (a device for transmitting and receiving data via the telephone) communication software and a telephone is capable of communicating with a mainframe and accessing its data base. Once a user is connected to a mainframe, he gains access by using a combination of the correct passwords and codes.

. Passwords and codes are supposed to be kept secret but sometimes lax security enables them to be attained quite easily.

- For example, in one case the unauthorized users learned the codes and passwords by getting computer runs out of the firm's dumpster; the codes and passwords were clearly printed on these runs.

- Another method used by unauthorized users is simply to "hack" it out; that is, to simply try different words and combinations until they hit upon the correct one. People who do this are known as "hackers."

4. What kind of crimes are committed by people who illegally gain entry to computer systems?

. Trespassing -- Gaining illegal access to a data base is as wrong as gaining illegal access to someone's property. It is trespassing--being where you don't belong.

- Violation of privacy -- Each citizen in the U.S. has a constitutional guarantee to privacy. Unauthorized users who gain access to data bases to simply look around, "tourists" as they are called, are violating the right to privacy of the people whose data they illegally review.

- Vandalism -- "Crashers", unauthorized users who deliberately destroy data base records, are guilty of vandalism.

- Larceny and forgery -- Unauthorized users who gain access to a system and profit by changing data may be guilty of both larceny and forgery.

5. What other kinds of problems have developed as a result of the increase in the number of data bases?

    - Because there is so much information available on each one of us, there is a resultant loss of privacy. By accessing different data bases it is legally possible to find out information on someone's health, educational achievement, credit history and even buying habits. Because this information is so easily available, people to some degree feel that part of their privacy has been taken from them.

Summary:

1. What do data bases contain?

2. How can computers infringe on our personal lives?

3. How do people use computers for illegal purposes?

Assignments:

1. Attempt to do some values clarification with the class. Have students do some role playing. Give volunteers roles to play: the hacker, the tourist, the crusher, the victim. Have students interact in these roles and have other class members respond to the attitudes of the role players.

2. Write a short story entitled, "The Boy Who Never Was." The basic plot is that no one, including his family and friends, recognizes the boy because his name and other data about him has mysteriously disappeared from all databases.

3. Teach the criminal justice system by having a mock trial of a "hacker." Students could be chosen to represent the various court officials and expert witnesses such as "tourists" and "crashers."

4. Create "WANTED" posters. Each poster should have a description of a computer-related criminal behavior to be on the lookout for.

5. Have students make a list of all the data bases they think their name may appear on. Expand the list to include the data bases that their parents' names may appear on.

TOPIC 13 - A

Computers in the News

Performance Objectives - Students will be able to:

- Identify current developments in the computer field.
- Describe the impact of computers on different fields.

Vocabulary: telecommunications, modem, hackers, hard copy, electronic mail, pacemaker

Motivation:

- Bring in a copy of a daily newspaper. Show that newspapers are usually divided into sections.
- Elicit the names of some of these sections and record these on the board.
- As the lesson develops, note how computers are having an impact on a particular field.

Aim:

Development:

1. Present the following items and then discuss the "Think It Through" questions at the end of the section.
   - The ability of computers to transmit and receive data is changing the way business is conducted in many offices.
     - To telecommunicate, what is needed is a personal computer, a modem, communications software and a telephone. Equipped with this, a business can send and receive electronic mail, for example, sales reports, charts, graphs, and memos.
     - Because of this ability, the significance of the "main office" is to some extent diminished. It may no longer be necessary to send all personnel to the main office to review data; it could simply be transmitted to them for review and their responses transmitted back.
   - Computers contributed by IBM and AT&T added greatly to the smooth operation of the 1984 Summer Olympics.
     - They helped facilitate the rescheduling problems created when the Soviet Union announced its boycott of the games. The thousands of athletes and hotel rooms, as well as the millions of dollars in ticket sales were all kept track of by computers.
     - Computers were even involved in the sporting events themselves as they were used to tally and display scores and do statistical analysis as well. Athletes were able to scout their competition by using the computer's file on the athletes.

  - AT&T operated an Electronic Message System which answered and delivered thousands of questions and messages, as well as transmitted thousands of stories each day of the Olympics.

- The F.B.I. has begun to get involved in the abuse of computer networks by hackers (people who illegally enter computer systems).

  - The F.B.I. held a series of coordinated raids in 13 cities in the U.S., seizing disks, hard copies and other evidence indicating unlawful entry into computer systems. Although many hackers try not to cause any damage when they enter a system, the potential for harm was illustrated when a group of users illegally entered the computer systems of the Defense Department and a system monitoring the medication given to cancer victims in New York.

  - As a result of this and other such episodes, the U.S. Congress is currently holding committee hearings in an effort to enact legislation to stop these abuses.

- Educational research has become much easier.

  - The "Academic American" by Grolier, an electronic encyclopedia, is currently available through the Dow Jones News/Bibliographic Retrieval Service and can be accessed through your personal computer with a telephone and modem. Grolier has also recently announced an encyclopedia in compact disk format.

  - To use the encyclopedia once it has been accessed by your computer, you simply provide the topic and you are given a choice of articles on the subject. Once you have made your selection, it will be displayed on the screen and if you have a printer a copy can be printed as well.

- Although computer industry experts predicted that 1984 would be a banner year for home computer sales, the industry experienced its worst slump.

  - Analysts attribute the unsatisfactory sales record to consumer confusion as what to do with a machine once they have it home. They point out that video games are no longer in vogue and most people would rather keep a recipe book on index cards than a computer.

  - The lower-priced home computers were worst hit by this slackened demand. A number of home computer manufacturers, including Texas Instruments, Timex, Coleco, and Mattel simply stopped producing their lines. Others, such as Atari and Commodore experienced severe financial troubles as well.

- Scientists have developed what they refer to as a "smart heart"

-pacemakers that are operated by tiny microcomputers.

- These devices which are about the size of a matchbox and cost less than $5000 are powered by long-life batteries. When implanted under the patient's skin, they can broadcast and be interrogated about the heart's performance. They can even be re-programmed within the patient's body to meet current needs of the patient.

Summary:

1. If transmitting and receiving data is possible with personal computers, what kind of impact will this have upon work as we know it? What groups of people will benefit the most from this technology?

2. Why were computers more greatly utilized in this Olympics than any other? Why was it appropiate that an Olympics held in Southern California should make such great use of computer technology?

3. Why should entering a computer system without permission be illegal? What kinds of damage could occur?

4. How can computers change the way we think of libraries? How will it effect certain groups of people who, because they are handicapped, cannot easily take advantage of our public libraries?

5. As a future computer consumer what warning is there in the fact that many manufacturers simply stopped producing their computer?

6. What are some other ways computers can be used to help doctors and other medical professionals?

Assignments:

1. Produce a computer news program with anchormen and feature reporters. One group of students can be assigned to produce the news program every two weeks. Use daily newspapers and weekly news magazines for information.

2. Create a computer newspaper complete with different sections indicating the different applications of computer technology. If applicable, include a classified section in which computer hardware and software could be sold and help wanted ads could appear.

3. Create a computer news bulletin board in the classroom. Find interesting articles and stories about computers which could be displayed.

4. If your school uses the public address system for morning announcements your students might wish to produce a "Computer News Note"--a brief story about computers and their application which would be presented each morning on the public address system.

TOPIC 14

Computer Etiquette - Do's and Don'ts

Performance Objectives - Students will be able to:

- Identify inappropriate behavior when working with microcomputers.
- Identify acceptable practices when using microcomputers.

Vocabulary: volatile, local area network, interfaces

Motivation:

- Initiate the lesson via a discussion of good manners in general, at home, at school, at a sporting event or at play.
- Note that each situation has its own definition as to what is acceptable and unacceptable.
  - It follows then that when working with computers certain behavior is unacceptable.

Development:

1. The keyboard is the primary input device on most computers and as such it is expected to get a great deal of use. This does not mean abuse.
   - When keying in programs or data, remember that under each key is an electrical contact. The more use it gets, the sooner it wears out.
     - It, therefore, makes little sense to depress keys over and over just to see the letters displayed on the screen or to try to challenge the computer by pressing down many keys at the same time.
     - Both practices are cause undue wear and tear and are clear signals that the user is not ready to operate a computer.
2. Since keyboards depend on making electrical contacts, care should be taken that nothing interferes with these contacts.
   - Food crumbs and spilled drinks could very well interfere with the electrical contacts. For this reason, no food or drink should be permitted near the computers.
3. Once set up, computers really should not be moved around, pushed or pulled.
   - The computer may be part of a "local area network" (computers connected together for transmitting and receiving data) and movement may disconnect interfaces (connections), disrupt the electrical source and wipe out the memory or, worst of all, fall off the table or desk.

4. The computer's memory is volatile. It can be erased easily and all that was stored there cannot be retrieved.

   - Be careful not to turn off the computer, interrupt the electrical source or type in "NEW" until the previous user has saved what's in memory to an outside storage device (disk or tape).

Summary:

1. Why should the computer's keyboard be treated with care?
2. Can the computer be moved around to make it easier to use?
3. Why should computer users be careful with the memory?

Assignments:

1. Make classroom signs to encourage acceptable behavior around computers.
2. Write an essay entitled: "How To Prevent Computer Breakdowns."
3. Construct a bulletin board on the theme "Computer Do's and Don'ts."
4. Develop a user's guidebook entitled, "The A B C 's of Using the Classroom Computer." For example, Always Ask Before Turning Off the Computer; Banging the Keyboard is Not Allowed; Care is a Must When Using the Computer.

TOPIC 15-A

Computer Futures

Performance Objectives - Students will be able to:

- Identify future advances in hardware.
- Identify projected developments in artificial intelligence.
- Identify advances in the input process.
- Explain what impact computers may have upon work as we know it.

Vocabulary: LSI, VLSI, ENIAC, input, output, process, scanner, OCR, artificial intelligence, telecommunicate, modem, data base

Motivation:

- Most students are probably aware of the famous Star Wars robots R2D2 and 3CPO. If possible, display a picture of them.

- Initiate a discussion as to what made these robots exceptional. Discuss the possibility of actually producing robots with these features.

Aim:

Development:

1. Since the late 1940's, the history of computers has been one in which they have gotten smaller, faster and cheaper.

    - The first generation computers used vacuum tubes to conduct the flow of electricity and were quite large. ENIAC, the first digital computer, had thousands of vacuum tubes and filled a large room.

    - With the development of transistors, integrated circuits and integrated circuit chips, computers took up less room and, because the electricity flowed better, calculations were accomplished much more quickly.

2. Currently, we are in the fourth computer generation.

    - Computers today are characterized by microchips. This process of large scale integration (LSI) allows hundreds and thousands of electrical circuits to be etched on a tiny silicon chip.

    - Fifth generation computers will be known for very large scale integration (VLSI). Each chip will contain a number of microchips. In other words, there will be more circuits crammed into a smaller area.

    - Where first generation machines did tens of thousands of calculations per second, fourth generation machines do hundreds of thousands of calculations per second.

3. The computing process has three distinct parts. They are: input, the entering of data; process, the manipulation of the data; and output, the displaying of results.

   - Most technological advances in computers have concentrated on the process portion. Each new microprocessor becomes smaller and more capable of processing data at increasingly greater speeds. Each new computer model boasts a larger memory capacity, allowing for more powerful programs and larger applications.

   - On the other hand, the input process has not improved as much. In fact, the primary input devices--disk, tape and keyboard--have not really kept pace with the increased processing speeds. The net result is that the input phase in a real sense presents a roadblock to realizing the full potential of better processors.

     - One improvement could be the increased use of optical character scanners (OCR). These are devices that can recognize characters (letters) on a printed page and input them to the computer. Unfortunately, the current crop of scanners are inaccurate and limited to a few typefaces.

     - Bar code readers, however, are becoming quite common in stores where UPC product codes are read and inputted to a cash register.

4. The most ambitious efforts in the data input field are in the field of voice recognition, the ability of the computer to recognize speech as commands. Two avenues of development are currently being followed.

   - The first is speaker-dependent recognition. This limits the number of users, but allows for a large vocabulary. The second area is speaker-independent. The number of speakers is increased, but the number of recognizable words is dramatically reduced.

   - Much of the difficulty in producing an effective system lies in the fact that hearing is not just an auditory function. Speech recognition systems also have to deal with enabling computers to discern between homonyms, differences in speech patterns, and meaning from the context of a sentence.

5. Each day brings the realization of artificial intelligence closer.

   - Computers have the ability to parallel and surpass the human brain in remembering data, performing mathematical operations and making logical decisions. However, they lack the ability to draw inferences, to apply past knowledge to a present situation and provide a prediction, and to distinguish between literal and figurative language.

   - Computer scientists, have, with some degree of success, been able to program computers to overcome some of these shortcomings. For example, the following software packages are in some senses "intelligent."

. Prospector

- This software makes predictions about ore deposits in certain areas based upon data presented to it.

. Internist

- This program makes a medical diagnosis based upon data presented by the doctor concerning patient's symptoms and tests. As the data begins to point in one direction the software probes deeper in this area and finally make a diagnosis.

. Frump

- This software writes a one-sentence summary of news articles in Chinese, Spanish, and English.

. Every day computers are being developed to parallel human thought. Thus, in a true sense, computers are becoming more and more intelligent.

6. The greatest future development lies in the application of the technology.

   . Modems currently allow anyone to connect to any one of a number of data bases (a systematic electronic collection of information). Through these data bases it is possible to bank, get mail, learn skills, research a topic, check stocks and shop.

   . Possibly more revolutionary will be the impact on how people work and live. As each day goes by, more and more people are working at home on their microcomputers and communicating with their firm's mainframe with a modem and the telephone lines. Telecommuting, the process of working at home and transmitting data over the telephone lines, may eventually become the way most people work. If it does, it will change family life, traffic patterns, energy usage and rents in office buildings.

Summary:

1. How are computers changing today?

2. Why does the increase in processing present input challenges for users?

3. If computers can understand human voice, do they in some sense think?

4. How will future computer applications change our society?

Assignments:

1. Discuss where computers are being used in the community. Develop a list of banks, supermarkets, schools, hospitals, etc. Visit some of these places to observe how input is accomplished, whether through scanners, keyboards or other devices.

2. Write a short story entitled "The Smart Computer." Assign the computer as much intelligence as desired. Afterwards ask student volunteers to read stories to the class and discuss the probability of certain abilities.

3. Practice business writing skills by writing to an information utility to secure information about its services. Write to: The Source, Source Telecomputing Corp., 1616 Anderson Road, McLean, Virginia 22102 or Compuserve, Personal Computing Division, 5000 Arlington Centre Blvd. Columbus, Ohio 43220.

4. Discuss telecommuting. How would this change many aspects of life? Make a list of jobs and rate each job's chances as a candidate for telecommuting. Jobs could be rated as possible, impossible or somewhat possible.

5. Construct a diorama illustrating the ways computers will be employed in homes of the future.

TOPIC 16-A

Types of Software

Performance Objectives - Students will be able to:

- Define software.
- Identify the purpose of software.
- Describe two different types of software.
- Describe at least four uses of software.

Vocabulary - New: software, firmware, courseware, canned or packaged software, programs, computer languages.
Review: Hardware, computer, input devices, output devices, memory devices, disks, tapes, cartridges, chips, cassette players, disk drives, printers, modems, paddles and joysticks

Motivation:

- What are some of the physical characteristics of a human being?
- What are the other "non-visible" qualities that make an organism a "human being"?
- What are some of the physical components of a computer?
  - Disk drive, printer, computer keyboard, modem, monitor.
- Hardware is analogous to the physical, visible characteristics of a human being, and software is analogous to those non-physical, non-visible characteristics.

Aim: What are the different types and uses of software?

Development:

1. Define software.

   - Software consists of the non-physical, non-visible components of the computer system. The programs that give the computer commands, instructions, and directions for doing a specific task are considered software. Software controls the physical or hardware components of a computer system.
   - Software comes on disks, tapes, cartridges, or chips.
   - It is important to remember that software written for one computer model is usually incompatible with another model. Hardware and software must be compatible.

- Note: Have samples of cassettes, disks, and/or cartridges available to show the containers used for storing software. Remember, although these modes are themselves physical objects, the information on these physical objects is the software.

2. There are two basic types of software and applications software.

   - Systems software (occasionally called firmware) is provided by the manufacturer to take care of computer maintenance tasks. Its job is to run the hardware (i.e., the video screen, printer, memory) and provide simple file management tools.

   - Applications software, which includes courseware, are programs that perform a particular function. For example, an addition program teaches children how to add, and a word processor manipulates text.

3. There are diverse uses of applications software. These include educational, business, personal, and recreational.

   - The uses of educational applications software include: learning from computers, learning with computers, learning about learning with computers, managing learning with computers, and learning about computers.

     - Educational Applications Software

   - Learning From Computers

     - Drill and practice and tutorials are programs that highlight learning from computers. Drill and practice programs assume the student has already been exposed to the content to be presented, whereas tutorials attempt to teach new concepts and information. Computer-Assisted Instruction (CAI) software includes both.

     - A simple drill and practice program written in BASIC can be found in Goldberg & Sherwood, 1983, pp. 157-159. Run it or one of your own creations to allow the students to observe an example.

     - An example of a tutorial is "Luna," a dialogue developed by Alfred Bork, University of California, Irvine.

   - Learning With Computers

     - Software that allows us to learn with computers includes simulations and games. In simulations, the computer presents us with a problem modeled after a real-world situation. With a simulation, we can make decisions based upon the data given and the consequences of previous decisions. On the other hand, a computer game is usually modeled on a fantasy situation.

     - A sample of a simulation is "Sell Lemonade," available from the Minnesota Educational Computing Consortium [MECC].

  - A sample of a learning game is "Hurkle," also available from MECC.

- Learning About Thinking With Computers

  - Software that helps us learn about thinking is not as plentiful as the drill and practice kind. It helps us to develop new patterns of thinking logically and analytically. (A sample of this would be Logo, available in many different versions.)

- Managing Learning With Computers

  - Paper work and administrative duties can be managed efficiently by using either commercially available or personally produced programs. There is software available for recording students' attendance and marks, finding the average of a set of grades, and generating test questions.

- Learning About Computers

  - In general, computer literacy, which includes computer awareness, refers to learning about computers. Most courses deal with the components of a computer system (for example, central processing unit, monitor, disk drive), the basic operation of a computer using the keyboard, voice or touch capabilities, computer languages, the impact of computers on society in general and on particular occupations both now and in the future, the use of pre-packaged software, and an introduction to actual programming.

  - A number of microcomputer and software manufacturers produce software that can be classified as learning about computers.

  - Non-educational Applications Software.

- Video games (PacMan).

- Word processing (Bank Street Writer).

- Personal investment analysis (Managing Your Money).

- Personal management and information (Personal Fitness Weight Control Program).

- Information and communication (Telelink I) allows for link up to a variety of information services such as The Wall Street Journal, and Dow Jones News.

  - Small business applications (Tax Preparation System, Personal Filing System).

- Note: Have many types of software to display so that the ideas presented can be made to be meaningful.

Applications:

1. Match each term in column A with the phrase that describes it in column B.

Column A

1. software
2. hardware
3. firmware
4. simulations
5. drill and practice
6. canned programs
7. Logo

Column B

a. A computer language that can help us learn about thinking with thinking with computers
b. software designed to learn with computers
c. The non-physical components of a of a computer system, including computer languages
d. Packaged programs commercially available for instructional or recreational use
e. Software designed to learn from computers
f. Systems software provided by the computer manufacturer
g. The physical components of a computer system

2. Arrange field trips to either a college or company that regularly uses computers so that the students can observe how software and hardware assist in the everyday tasks of institutions.

Summary:

1. What is software?

2. What are some of the purposes of software?

3. What are some of the different types of software?

4. What are some of the uses of software?

Assignment:

1. After participating in a field trip to either a college or business that employs computers, write a short description of how the computer helps that institution to function and complete necessary business.

Assignment for More Advanced Students:

1. Present students with three to five pieces of software that can be classified as learning from computers, learning with computers, learning about thinking with computers, and ask them to identify the category in which each belongs.

TOPIC 17

Following Directions

Performance Objectives - Students will be able to:

- Recognize the importance of following directions carefully.
- Execute a series of directions.
- Predict the outcome of a series of directions.
- Determine whether the series of directions is complete and logical.

Vocabulary: logical, sequence, ordering

Motivation:

- Write the directions to walk from school to the nearest candy store for a friend who is new to the school. Include the approximate number of steps and when and how much to turn right or left.

- Allow students to exchange and read each other's papers. Discussion and comparison of students' directions should follow.

- Comment on the order of executing the directions.

- What would happen if the order were changed?

Aim: How do we follow a series of directions?

Development:

1. Why is it important to give clear and precise directions?

2. Where do we find directions and instructions for doing things in our everyday lives? (recipes; putting models together; taking tests; assembling equipment and toys such as bikes and computers; directions to go to and find a certain location.)

3. What would happen if we did not read the directions in order or listen to directions carefully?

4. What would happen if the directions were not clear or were incomplete?

5. We are able to follow directions that we understand. Look at the two sets of directions. Which package has clearer instructions?

Brand A	Brand B
Add water and eggs. Bake in oven.	Add 1 cup of water and 3 eggs. Bake in oven at 375 degrees for 45 minutes.

- Brand B has better directions because they are complete.
- Since we can think about and interpret information, it is possible that we could make a good cake using either recipe.
- Computers, however, cannot think, so we must be sure that the directions we give to the computer are clear and precise.

6. Let's look at the following directions for using a telephone. Is there anything wrong with them?
    - Listen for a ringing sound.
    - If busy or no answer, return receiver and try again later.
    - Listen for dial tone.
    - Await pick up at other end, when someone speaks, answer.
    - Pick up receiver.
    - Dial or press numbers in order from left to right.
    - Put the above directions in proper sequence. Once the sequence is in proper order, discuss the logic.
7. How to follow directions:
    - Focus full attention on the directions in whatever form you find them.
    - At the end of a difficult step, stop, reflect, and reread.
    - Make sure you understand the meaning of any new words.
    - Examine any accompanying photographs and diagrams.
    - Try to handle the directions on your own without any help.
    - Try to recognize any errors you might make in interpreting the directions.

Applications:

1. Collect a sample of recipes or other series of directions. Paste the directions on one side of a piece of construction paper and the title of the recipe on the other side. Ask students to predict what the outcome of the series of directions will be.
2. Collect a sample of recipes or other series of directions. Rearrange the order and ask students to put the series in proper sequential, logical order.

3. Incorporate this lesson with an art, science, and/or mathematics lesson where students must follow directions to create an art project, complete a science experiment, or solve a problem.

Summary:

1. Why is it important to follow directions carefully?

2. When following a series of directions, what are some of the guidelines we should keep in mind?

3. How can we tell when information in a series of directions is not presented in proper sequence?

4. How do we predict the outcome of a series of directions?

Assignment:

1. Find a recipe or other series of directions. Paste the directions on the front of a piece of construction paper and the title on the back of the paper. The next day, ask students to exchange their directions and have their neighbors predict the outcome.

TOPIC 18

Practice Keyboarding Skills on the Apple

Performance Objectives - Students will be able to:

- Use the alphabet and number keys.
- Use selected special function keys.
- Demonstrate how to move the cursor around the screen.

Vocabulary: cursor, space bar, delete, home, scrolls, functions

Motivation:

- Show a typewriter and a computer.

- How are these two machines similar? different?

Aim: How do we use the special control keys on the Apple keyboard?

Development:

1. Note that computers and typewriters both have keyboards, but the computer has special keys to do special jobs or functions.

   - Both have a specially-shaped key called a space bar. Ask "What happens when we press the space bar?" (rectangle of light moved forward one space.)

   - State that this rectangle of light is called a cursor. Ask "What does this cursor show you?" Press space bar several times.

     - Note: cursor comes from the Latin word meaning runner.

2. Find the key marked ">". It is called the cursor right arrow key. Press it once. What happened? (The cursor moved to the right.)

   - Find the key marked "<". It is called the left arrow key. Press this key and the shift key. What happened? (The cursor moved backward one space.)

   - Type your name. Press the < key one time and press the space bar. What happened?

3. Type HOME and press return. What happened? (Cursor moved to the top and the screen disappeared. You told the computer to erase the screen.)

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
Space - bar	Moves cursor ahead one space
> right arrow key	Moves cursor ahead
< left arrow key	Moves the cursor back one space
HOME	Returns cursor to top of screen and erases what is on the screen.

4. Type your name. Look at the keyboard and find the ESC key. Press ESC and then press the "M" key. What happened? (The cursor moved down a line.)
   - Press ESC and then press the "I" key. What happened? The cursor moved up one line.
   - Press ESC and then press the "J" key. What happened? The cursor moved left one space.
   - Press ESC and then press the "K" key. What happened? The cursor moved right one line.
5. Type your name. Press the return key 24 times. What happened?
   - Note that the screen has only 24 lines and that as you add more lines the top line scrolls (moves upward) and off the screen.

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
ESC + I	Cursor moves up one line.
ESC + M	Cursor moves down one line.
ESC + J	Cursor moves left one space.
ESC + K	Cursor moves right one space.

Applications:

1. Practice using the cursor keys to correct, add to or subtract from, a paragraph.
2. Copy a poem or short story using upper and lower case.

Summary:

1. How are computer keyboards similar to typewriter keyboards?
2. How are computer keyboards different from typewriter keyboards?
3. What is the purpose of each special function key?
4. What do we call the blinking light? What is its function?

Assignments:

1. Make a chart or bulletin board showing the special function keys and their use.
2. Make a model of the computer keyboard showing the special function keys.

TOPIC 19

Practice Keyboarding Skills on the Commodore Pet

Performance Objectives - Students will be able to:

- Use the alphabet and number keys.
- Use selected special function keys.
- Demonstrate how to move the cursor around the screen.

Vocabulary: cursor, space bar, delete, home, scrolls, functions

Motivation:

- Show a typewriter and a computer.

- "How are these two machines similar? different?"

Aim: How do we use the special control keys on the Commodore Pet keyboard?

Development:

1. Point out that computers and typewriters both have keyboards, but the computer has special keys to do special jobs or functions. Both have a specially-shaped key called a space bar.

   - What happens when the space bar is pressed? (Rectangle of light moved forward one space.)

     - State that this rectangle of light is called a cursor.

   - What does this cursor show? Point out that the word comes from the Latin word meaning "runner." Press space bar several times.

2. Recall that the computer can do more than a typewriter.

   - Find the key marked CRSR ---->. It is called the cursor right arrow key. Press it once. What happened? The cursor moved to the right.

   - Find the key marked CRSR <----. It is called the left arrow key. Press this key and the Shift Key. What happened? The cursor moved backward one space.

   - Type your name.

   - Press the CRSR <---- key one time and press the space bar. What happened?

   - Find the DEL key. Press it once. What happened? How is this key different from the CRSR key? What is this key used for? DEL stands for delete or remove.

3. Find the HOME key. Press it once. What happened? Cursor moved to the top and the screen stayed the same.

4. Press the key marked "CLR" and the shift key. What happened? The screen display disappeared, and the cursor moved to the top line.

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
Space - bar	Moves cursor ahead one space.
CRSR - Right Arrow key	Moves cursor ahead.
DEL	Backspaces and deletes one letter.
HOME	Returns cursor to top of screen and keeps what is on the screen.
CLR + SHIFT	Returns cursor to top of screen and erases the screen.

5. Look at the keyboard and find the CRSR DOWN key. What is the function of this key? Press it.

6. Type your name.

   - Press the CRSR DOWN key 25 times. What happened? Note that the screen has only 25 lines and that as you add more lines the top line scrolls (moves upward) and off the screen.

   - Press the return key twice. What happened? Cursor moved down two lines.

7. Type your name. Press the SHIFT key and the CRSR UP key. What happened? Cursor moved up two lines.

8. Press HOME and then press the "RVS" key. Type your name. What happened? Letters printed in reverse. What does RVS mean?

   - Press SHIFT + CLR and then press OFF. Type your name. What happened? Reverse function turned off.

9. Add the following to the previous chart.

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
SHIFT + CRSR UP	Moves cursor up.
RVS	Prints in reverse.
OFF	Turns off reverse; prints normally.

Applications:

1. Practice using the cursor keys to correct, add to, or subtract from a paragraph.
2. Have a student copy a poem or short story using upper and lower case.

Summary:

1. How are computer keyboards similar to typewriter keyboards?
2. How are computer keyboards different from typewriter keyboards?
3. What is the purpose of each special function key?
4. What do we call the blinking light? What is its function?

Assignments:

1. Make a chart or bulletin board showing the special function keys and their use.
2. Make a model of the computer keyboard showing the special function keys.

TOPIC 20

Practice Keyboarding Skills on the TRS-80

Performance Objectives - Students will be able to:

- Use the alphabet and number keys.
- Use selected special function keys.
- Demonstrate how to move the cursor around the screen.

Vocabulary: cursor, space bar, clear, scrolls, functions

Motivation:

- Show a typewriter and a computer.

- "How are these two machines similar? different?"

Aim: How do we use the special control keys on the TRS-80 keyboard?

Development:

1. Note that computers and typewriters both have keyboards, but the computer has special keys to do special jobs or functions. Both have a specially-shaped key called a space bar.

   - What happens when the space bar is pressed? (Rectangle of light moves forward one space.)

   - State that this rectangle of light is called a cursor. Ask, "What does this cursor show you?" Point out that the word comes from the Latin word meaning "runner." Press space bar several times.

2. Recall that the computer can do more than a typewriter.

   - Find the key marked ---->. It is called the right arrow key. Press it once. What happened? The cursor moved ahead 8 spaces.

   - Find the key marked <----. It is called the left arrow key. Press it once. What happened? The cursor moved backward one space.

   - Type your name. Press the <---- key one time and press the space bar. What happened? What is the <---- key used for?

3. Press the key marked "CLEAR." What happened? The screen display disappeared, and the cursor moved to the top line.

   - Type this sentence, "I am learning to use a computer." Now hold down the "SHIFT" key and at the same time press the <---- key. What happened? (The entire line was erased, and the cursor returned.)

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
Space - bar	Moves cursor ahead one space.
Clear	Clears screen, returns cursor.
----> Right Arrow key	Moves cursor 8 spaces forward.
<---- Left Arrow key	Backspaces and erases one space.
Shift + <----	Erases entire line, returns cursor.

4. NOTE: For the rest of this lesson the "**" symbol is used to represent the down arrow. Look at the keyboard and find the "**" key. What is the function of this key? Try pressing it.

   . Type your name. Press the "**" key 16 times. What happened?

   . Note that the screen has only 16 lines and that as you add more lines the top line scrolls (moves upward) and off the screen.

5. Press Shift + ----> key and type your name. What happened? Letters appeared to be double width.

   . Press CLEAR and type your name. What happened? The display cleared and returned to the top of the page. Letters returned to normal size.

6. Hold down the shift key and press 0. Type your name. What happened? All of the letters are lowercase letters. How can we make the first letter of your name a capital letter? (The shift key plus the letter)

   . Hold down the shift key and press 0. Type your name. What happened? You returned to all capital letters (uppercase).

7. Add the following to the previous chart.

FUNCTION KEY CHART

KEY	FUNCTION (JOB) PERFORMED
(**) - down arrow	Moves cursor to beginning of next line.
Shift + ---->	Creates double-width letters.
Clear	Returns to normal-width letters.
Shift + 0 (zero)	Switches upper-case to lower-case and vice-versa.

Applications:

1. Practice using the cursor keys to correct, add to, or subtract from a paragraph.
2. Have a student copy a poem or short paragraph using upper- and lowercase.

Summary:

1. How are computer keyboards similar to typewriter keyboards?
2. How are computer keyboards different from typewriter keyboards?
3. What is the purpose of each special function key?
4. What do we call the blinking light? What is its function?

Assignments:

1. Make a chart or bulletin board showing the special function keys and their use.
2. Make a model of the computer keyboard showing the special function keys.

TOPIC 21 - E

Verbal Direction Game

Performance Objectives - Students will be able to:

- Follow verbal directions.
- Evaluate contradictions in directions.
- Explain a sequence of ideas.

Vocabulary: start, stop, spot, direction, turn

Motivation:

- Play a game of "Simon Says" with the class.

Aim: To sequentially follow directions

Development:

1. Pick a team of students to be "Searchers." They leave the room and select one member to be blindfolded.
2. The rest of the class decides on a spot on the floor and marks it with tape.
3. The Searcher team returns and attempts to give verbal directions to the blindfolded member so as to reach the designated spot.
4. After each command the rest of the class responds to the direction by saying "warmer" or "cooler," depending on whether the student is moving closer or further from the spot.
5. After the team successfully has the member reach the spot, a new team of Searchers is selected and the game is started over.
6. A record of how many verbal directions each team used to find the spot is kept. The winning team is the one that reaches the spot with the least directions.

Summary:

1. What made the team win? (less directions)
2. What did the blindfolded Searcher need to do? (listen carefully)

Application:

1. Discuss, develop and demonstrate with the class the shortest possible sequence of directions a player could be given.

Assignment:

1. Divide the students to play Tic-Tac-Toe. Members are either on the "X" team or the "O" team. Allow them to play several games in which they are required to verbalize the placement of marks. Students can designate the placement by such directions as left, center, right, and top, middle, bottom.

TOPIC 21 - M

Verbal Direction Game

Performance Objectives - Students will be able to:

- . Follow verbal directions.
- . Record sequential information.
- . Solve oral problems through listening for specific data or clues.

Vocabulary: direction

Motivation:

- . Instruct class to listen, record and repeat specific directions in walking to a nearby neighborhood site, such as store, park, library, etc.

- . Call on students to retell the directions until they are retold exactly.

Aim: To discriminate and use verbal directions.

Background:

1. Discuss the need to develop skills in following directions.

2. Much of what we learn, we acquire by listening to directions (at home, at school, at play, etc.)

3. Following directions involves: hearing, understanding, evaluating, responding.

Development:

1. Play "Do As I Say." Give three or more directions in sequence and choose a student to carry them out. For example: Go to the back of the room, turn around three times, pick up a book from the table, give it to John, walk over to the window, count to ten, and then go and sit in your seat.

2. Depending upon the students' ability and age, increase the complexity of the directions.

3. Allow the student to finish the sequence even though a mistake may have been made.

4. Elicit from the rest of the students an evaluation of whether or not the student completed the tasks correctly. When it is agreed that a student has made an error, another is chosen.

5. Ask students to similarly create and write their own directions for "Do As I Say." Select students to verbally direct others to follow this sequence of tasks.

6. Have students follow specific instructions for cutting and folding pages.

7. Ask three students to display their finished products. Ask students to evaluate if it is the result of the directions.

Applications:

1. Have students work in pairs. Instruct each to plan a shape, with the monitor screen off. Taking turns, each verbally directs the other to have the turtle make the shape step by step.

2. When the directions are finished, they can turn on the monitor to see if the shape is correct.

Summary:

1. Why listen from the beginning?

2. Why listen carefully to each detail or step?

3. Why think instructions through as you carry them out?

Assignment:

1. Explain, in order, the sequence of movement during a fire drill.

TOPIC 21 - A

Verbal Direction Game

Performance Objectives - Students will be able to:

- Follow verbal directions in sequential order.
- Develop and convey sequence of directions.

Vocabulary: degree

Motivation:

- Show the face of a clock. Discuss how its circle is variously divided (12 hours, or 60 minutes, or 43,200 seconds)

Aim: To develop verbal directions in sequential order.

Development:

1. Draw a circle on the chalkboard. Explain that one complete turn of the circle requires 360 degrees.

2. Have a student demonstrate in front of the class by turning 360 degrees, 180 degrees and 90 degrees, while another student follows each corresponding movement on the chalkboard.

3. Ask all students who are seated to point to where 90 degrees to the right would be from where they are sitting. Then ask them to point to where 90 degrees to the left would be.

4. Explain that they will all practice together using directions along with turning degrees in a game.

5. If students have completed Topic 39 Early Experience, explain that this game is similar but will use degrees instead.

   - Select a team of students to be "Searchers." They leave the room and choose one member to be blindfolded. While waiting outside, the Searcher can practice turning in degrees and movement of steps while blindfolded.

   - The rest of the class marks an agreed upon spot on the floor with tape.

   - The Searchers return and attempt to give verbal directions to the blindfolded member so as to reach the designated spot.

     - They are limited to four types of command: right or left plus degrees, forward or back plus the number of steps.

     - For example, they can say, "right 45 degrees," or "forward three steps."

- After each command, the rest of the class responds by saying "warmer" or "cooler," depending upon whether the student is moving closer or further from the spot.

- After the team successfully has the member reach the spot, a new team of Searchers is selected and the game is started over with a new spot.

- A record of the verbal directions each team used to find the spot is made on the chalkboard. The winning team is the one that reaches the spot with the fewest directions.

Summary:

1. How many degrees equal a circle? a half-circle? a quarter-circle?

Assignment:

1. Draw the geometric shapes discussed and mark the proper number of degrees in each.

TOPIC 22 - E
Playing Turtle

Performance Objectives - Students will be able to:

- Physically move in reply to designated directions.
- Practice the four basic turtle movements: forward, back, left and right.
- Record instructions from turtle movements.

Vocabulary: turtle, left, right

Motivation:

- What is the slowest animal you know? (snail, turtle, sloth)

- Could you describe exactly how it moved if you watched it very closely? Why?

- Today we are going to talk to a turtle, and watch it move in our classroom. How is that possible? (Elicit that they will play like a turtle.)

Aim: How can we move like a turtle to follow directions?

Development:

1. Stand in the center of the room and instruct the class that you will be called "turtle."

2. Elicit simple movement commands to give the turtle (forward, back, turn left, turn right, steps, etc.)

3. Suggest that instead of just telling the turtle to turn left or right, they use the twelve numbers on a clock as commands for the turtle. Explain that the turtle will always look at a watch to be guided.

4. Have students direct the "turtle" (teacher) to a different part of the room. Follow instructions even if incorrect and elicit more exact commands.

5. Draw a shape on the board. Have students tell "turtle" to walk the shape in the classroom.

6. Select students to be the turtle and to follow the same procedure.

Applications:

1. Draw a picture showing a turtle moving on paper. You play turtle while a student gives you instructions to walk the design.

2. Give the commands to a student-turtle to walk out his or her initials.

3. With your pen, draw a line ten steps each, turn to three o'clock, go forward five steps, turn to nine o'clock, go forward five steps, turn to nine o'clock, go forward ten steps. What letter did you make? (U)

Summary:

1. What ways can our turtle move?

2. From your seats, which direction is three o'clock? Six o'clock? Twelve o'clock?

Assignment:

1. Have students record all their walking movements from the school entrance to the classroom seat.

TOPIC 22-A
Playing Turtle

Performance Objectives - Students will be able to:

- Exhibit an awareness of the concepts of body geometry.
- Identify the two turtle moves: FORWARD and BACK.
- Identify the two turtle turns: LEFT and RIGHT.
- Execute the four turtle commands while playing turtle.
- Issue directions for the four turtle commands while playing the role of turtle trainer.

Vocabulary: turtle, robot, trainer, commands, turns, moves, FORWARD, BACK, LEFT, RIGHT

Motivation:

- Discuss robots with the class.

- Elicit what a robot can do and list responses on the board. If students' suggestions do not include the need to program or instruct the robot to perform a function, introduce and expand this concept.

  - What is a robot?

  - What kinds of things can a robot do?

  - What makes the robot perform as it does?

Aim: How can we "play turtle?" How can we play the part of the turtle robot? How can we play the part of the turtle trainer?

Development:

1. Students should take turns pretending to be a special kind of robot, called a turtle, and being the turtle trainer, the person who gives the turtle commands.

2. Act as the first turtle trainer to set up a model. Choose a student to act the part of the turtle. The turtle can only "understand" four commands (directions). Two of these commands are moves: FORWARD and BACK. The other two are turns: LEFT and RIGHT.

3. Issue the command FORWARD several times so they can see the effect of a forward progression. The number of steps, how far forward, may also be used depending on the determined needs of the group.

4. Use the turtle to execute (carry out) the commands BACK, RIGHT and LEFT.

Applications:

1. Goal Game: Students take turns participating as turtle and turtle trainer. The object is to have the turtle reach a goal, a set endpoint. It could be a fixed endpoint, such as the classroom door or a certain bulletin board, or an object, such as a box or a book.

2. Group Games (adaptations of traditional children's games)

   . Mother May I?: group participation, one turn at a time.

   - Participants line up at a set distance from the student who has been chosen to be the leader or mother. The object of the game is to reach the leader first and take his/her place.

   - The leader issues a turtle command to the first participant in the line. The participant must then ask permission "Mother (Leader, Trainer, Father), may I?" If the leader gives permission, the participant follows the direction. The leader may opt to deny permission and issue a new command.

   - A participant who does not ask permission is sent back to the starting line. The game continues across the line with each participant in turn until someone reaches the leader or time runs out.

   . Simon Says: whole group participation

   - A leader, Simon, gives directions to the whole group at once. The object is to be the last remaining participant in the game, thus becoming the new leader.

   - This goal is accomplished by responding only when a command is preceded by "Simon says." Those who respond to commands not preceded by "Simon says" are eliminated from the game.

   - Any or all of the four turtle commands may be used.

Summary:

1. What are commands?

2. What does a turtle trainer do?

3. What commands do our turtle robots understand?

4. Which turtle commands are moves?

5. Which turtle commands are turns?

Assignment:

1. On an index card, write a sequence of five or more turtle commands.

   . The format depends on the students' abilities. At an early level, they could use the letters F, B, L, and R to represent commands. These cards can be used as a follow-up lesson.

Assignments for More Advanced Students:

1. On an index card, write a sequence of five or more turtle commands with inputs (number of steps for moves, number of degrees for turns).

   . These cards can be used in follow-up lessons.

2. Write a sequence of commands to direct a turtle to another part of the school building (the gymnasium or the main office).

TOPIC 23 - E

Editing Directions In a Verbal Game

Performance Objectives - Students will be able to:

- Follow verbal directions.
- Change directions in a verbal game.
- Predict the consequences of changing directions.

Vocabulary: direction, PacMan, ghost

Motivation:

- Draw a simple maze on the board. Have a student place a finger at the starting point. Call on the class to direct the student to trace the maze to the end. If a wrong direction is given, allow the movement to be made and elicit other suggestions to correct it.

Aim: How can we follow or change directions in a verbal game?

Development:

1. Discuss the fact that everyone makes mistakes and needs to do things over again or differently. (i.e., learning to ride a bicycle, spelling a word.)

2. Tell the class that they will see this as they play a game of PacMan.

3. Rearrange desks in a maze pattern. Form one team of five students called "Pacs" and another called "Ghosts."

4. Arrange the Ghosts in various spots throughout the maze.

5. One of the Pacs begins the course while the rest of the team verbally takes turns giving directions. The goal is to jointly get the Pac through the maze without getting near a Ghost. If a Ghost gets to touch the Pac at arm's length, the Pac is out.

6. Continue with other Pac team members and then reverse or change teams.

Summary:

1. Have students work together to design, draw and color their own PacMan games. Allow them to play the game in groups.

Assignment:

1. Have students create a set of directions for changing a flat tire.

TOPIC 23 - A

Walking a Maze

Performance Objectives - Students will be able to:

- . Understand the concept of a maze problem.
- . Devise simple strategies to solve a maze problem.
- . Experiment using various strategies.
- . Assess and modify strategies for maze problems.

Vocabulary: maze

Motivation:

- . Draw a simple maze on the chalkboard. Explain to the students that they are to "make believe" they are a mouse who needs to get the cheese on the other side. They have only 15 seconds to find it.

- . Ask for volunteers to point out their path within 15 seconds.

- . Continue until it is successfully done.

Aim: To use various strategies in a maze problem.

Development:

1. Have the students rearrange their desks and chairs to form a classroom maze.

2. While the rest of the class is setting up their chairs, one student starts walking from the back of the room through the maze to the teacher up front. Ask the rest of the class to count the turns.

3. Ask the student to reverse his/her steps and return to the starting point. Ask the rest of the students to count the total turns made again.

4. Discuss if and why the turns the student made were different each time.

5. Rearrange the desks differently. Have volunteers walk different mazes with any of the following variances:

    - . Two students walk the maze in a race.

    - . Students try walking backwards through the maze.

    - . Have the students stand up and extend their arms out as one of them walks through.

    - . Have a student attempt to walk through blindfolded.

Application:

1. Have students draw their own mazes and pass them to another student to solve.

Summary:

1. What is a maze?

2. How do you go about getting through it?

3. How is our neighborhood like a maze for us?

Assignment:

1. Draw a diagram of the rooms of your house. Show how you walk from your front door to your bedroom.

TOPIC 24

Importance of Correct Sequence

Performance Objectives - Students will be able to:

- Recognize the relationship between a complete and correct sequence of events and the desired outcome.
- Demonstrate knowledge of a problem-solving approach using deductive thinking.

Vocabulary - New: Task, product, procedure, end.
Review: sequence, robot, recipe, goal, order

Materials: Loaf of bread, jar of jelly, jelly knife.

Motivation:

- Have students write simple directions for making a jelly sandwich.

- Discuss the directions and select one or two "recipes" to place on the board.

  - With students whose writing skills do not permit completion of the independent task, allow development of ideas orally and record the suggested steps on the board.

Aim: How do the steps we follow in a sequence change the outcome?

Development:

1. Understanding and defining the problem.

   - What is the task? (to make a jelly sandwich.)

   - Let's put it on the board this way to start our procedure (recipe, set of directions, list of instructions or commands):

```
to make.a.jelly.sandwich
```

2. Developing a procedure. (a plan to accomplish a task)

   - If possible, use a set of directions which will lead to an unintended product.

   - We have worked hard to write a step-by-step plan for the robot.

   - Now that we have stated our task goal, we can list our instructions underneath it. We must also let the robot know when we have reached the end of our instructions.

   - The following is an example of what may appear on the board after the instructions and end statement are added:

```
to make.a.jelly.sandwich
   place a piece of bread on a flat surface
   put the jelly on the bread
   put another piece of bread on top of the jelly
 end
```

   - Refer to this as a procedure.

3. Following the procedure.

   - A student who can carry out directions literally may be chosen to be the robot.

   - Using the procedure in step 2, an unexpected outcome may be that the robot puts the jar of jelly on the piece of bread.

     - This example clarifies the concept of an incomplete sequence.

   - Another example may indicate the results of an incorrect sequence.

4. Looking back on the procedure.

   - What happened?

   - Can you explain why our procedure did/did not accomplish our task?

   - If it did not accomplish our task as we predicted, how can we fix it?

   - Would you like to try it again with our changes or improvements?

Applications:

1. Collect directions, e.g., cake mix, pudding. Evaluate the directions and find possible ways to improve on them.

2. Utilize the four-step approach to task completion for lessons in arts and crafts, science experiments, physical education skills, or mathematical topics.

3. Develop a task file.

   - Use recipes, mix directions, game directions or other appropriate set of directions.

   - Place individual steps of procedures on individual pieces of paper or index cards. Place all of the steps necessary for a specific procedure into an envelope which is clearly marked, for example: to.find.buried.treasure.

Summary:

1. What are the four steps we use in planning and carrying out a procedure?
2. What happens when we leave steps out of our sequence?
3. What happens when we put the steps to our procedure out of order?
4. What are some suggestions for making sure that our procedure does what we want it to do?

Assignment:

1. Display a picture of a simple house with a sun and one tree. Have the students write a procedure to.draw.house.picture. Exchange procedure with neighbor, follow the procedure and evaluate the finished product.

   . Younger students or those with limited writing skills could dictate steps to the teacher, who would then write the procedure on the board or chart paper and place pictographic cues next to each step.

Assignments for More Advanced Students:

1. Individually, or within a small group setting, write a procedure for playing a familiar game such as tic-tac-toe, hangman, checkers or other popular board games. Exchange directions, follow the procedure and evaluate it.

TOPIC 25

Geometric Shapes

Performance Objectives - Students will be able to:

- Define a circle, a square and a triangle.
- Draw geometric shapes.
- Recognize geometric shapes around them.

Vocabulary: shape, box, line, square, triangle

Aim: To learn to recognize basic geometric shapes in nature.

Motivation:

- Display a large piece of cloth that is decorated with various geometric shapes.
- Have students point to various different shapes on it.

Development:

1. Draw a circle, square and triangle on the board.
2. Ask students to explain how the shapes differ.
3. Elicit the name of each shape.
4. Ask students to point to something in the classroom that looks like each of these shapes.
5. Have them draw a scene that has a circle, square and triangle in it.
6. Allow students to narrate their scenes to the class.

Summary:

1. Review the identity of each shape.

Assignment:

1. Search through old magazines and cut out and paste in your books picture shapes of circles, squares, and triangles.

TOPIC 26 - E

Introduction to Logo

Performance Objectives - Students will be able to:

- Name the computer language they are learning.
- Identify the screenturtle.
- Explore the effects of Logo commands in the immediate mode.

Vocabulary - New: Logo, screenturtle, DRAW or CLEARSCREEN
Review: FORWARD, BACK, LEFT, RIGHT, cursor, Return or Enter

Motivation:

- Elicit the commands used in "playing turtle." Write FORWARD, BACK, LEFT, RIGHT on the board. Write F, B, L, R next to the appropriate words.
- These are some of the commands we need to know to meet another kind of turtle and to begin using a computer language called Logo.

Aim: How can we use the "playing turtle" commands on the computer?

Development:

1. Have the children meet the screenturtle by displaying a facsimile on the board or by typing the appropriate command at the keyboard and showing the turtle on the screen. Add the command necessary for meeting the screenturtle to the list on the board. DRAW or CLEARSCREEN, for example, may be the appropriate command and may be accessed by using D or C. This would depend on the specific instant or single-keystroke program being used.

2. What were the commands that we used when "playing turtle?" The screenturtle can follow the same commands. We use F and B to tell the turtle to move forward or back a certain number of steps. We use R or L to tell the turtle to turn left or right a certain number of degrees. Refer to the single letters as FORWARD 15 steps, BACK 15 steps, LEFT 90 degrees, RIGHT 90 degrees, or whatever number represents the actual input in the version that the children will be using.

3. What else do we have to know? We need to know how to erase our drawing so that the next person may start with a clean screen. Explain what command must be issued to clear the screen and home the turtle in the implementation of Logo that the students are using. The commands and letters used would normally be those used in step 1, above.

4. Demonstrate how the program works on the computer. The blinking cursor is waiting for us to press a key. The turtle only understands the letters we have talked about and have listed on the board. In a followup lesson, other commands may be added. Commands which control penup and pendown functions so that the turtle will move without leaving a trail are examples.

NOTE: Logo programs used for instant or single-keystroke responses usually work on a command-key principle. Certain letters have been chosen to represent commands and those are the only letters which will evoke a response. Other letters, those which do not represent a command, will be ignored. The following two steps, therefore, may be considered optional.

5. Suppose you were writing a word on a piece of paper and you wrote one letter that you didn't want. How would you get rid of that letter? We have a way of erasing a letter in Logo, too. Explain how to erase a letter or letters in the implementation of Logo the students are using.

6. What games do you know of in which you need permission before you can move? (Examples: Mother may I?, Simon Says) You have to give the turtle permission before it moves. You do that by pressing the return key (enter key).

Application:

1. Students work individually or in pairs exploring the effects of the Logo commands in immediate mode. Children working in pairs may approach the activity cooperatively contributing to the same end product. The chance-apiece method, each member of the pair taking a turn being the turtle driver and the turtle observer, may also be used.

Summary:

1. What is the name of the computer language we are learning?
2. What are the commands for turtle turns?
3. What are the commands for turtle moves?
4. What do we type in order to see the screenturtle?
5. What do we type to erase our drawing and make the turtle go back home?

Assignment:

1. Make a design using each of the four main Logo commands at least twice. After the design is finished, show how to erase the picture from the screen.

Assignments for More Advanced Students:

1. Explore the possibility of recreating familiar shapes using the screenturtle. Describe the picture orally or dictate a short description to the teacher, who can enter the phrase or sentence into a pupil log.

2. Task cards designed by the teacher, picturing simple line designs or shapes on oaktag or index cards, may be used as a challenge activity or with students who are having trouble deciding what they would like to draw.

TOPIC 26 - A

Introduction to Logo

Performance Objectives - Students will be able to:

- . Name the computer language they are learning.
- . Identify the screen turtle.
- . Use Logo commands for moves: FORWARD and BACK.
- . Use Logo commands for turns: LEFT and RIGHT.
- . Manage the housekeeping functions necessary for erasing text and clearing the graphics screen.
- . Use Spacebar.

Vocabulary - New: Logo, screenturtle, spacebar, DRAW or CLEARSCREEN.
Review: FORWARD, BACK, LEFT, RIGHT, cursor, RETERN or ENTER

Motivation:

. Elicit the commands used in "playing turtle" or ask what commands could be used in a marching activity.

- Write FORWARD, BACK, LEFT, RIGHT on the board.

. These are some of the commands we need to know to meet another kind of turtle and to begin using a computer language called Logo.

Aim: How can we use the "playing turtle" commands on the computer?

Development:

1. Introduce the screenturtle by displaying a facsimile on the board or by typing the appropriate command at the keyboard.

   . Add the command to the list on the board.

2. The screenturtle follows the same commands used when "playing turtle."

   . Simply tell the turtle how many steps to move forward or back. Press the spacebar. Then, tell it how many degrees to turn left or right.

   . Write an example on the board to demonstrate the format: command-space-number.

3. How do we know that one word is ending and another word beginning when we hear someone speak? (a pause between words) The turtle needs a space (pause) to know where the word ends and the number begins. Logo is fussy about having a space between a word and a number.

4. Suppose you were writing a word on a piece of paper and you wrote one letter that you didn't want. How would you get rid of that letter? We have a way of erasing a letter in Logo, too.

- Explain how to erase a letter or letters in the implementation of Logo the students are using.

5. What else do we have to know how to erase? We need to know how to erase our drawing so that the next person may start with a clean screen.

   - Explain what command must be issued to clear the screen and home the turtle in the implementation of Logo that the students are using.

6. What games need permission before you can move? (Mother may I?, Simon Says)

   - The turtle needs permission before it moves, too. Instead of saying "You may", you press the return key (enter key).

Application:

1. Students work individually or in pairs exploring the effects of the Logo commands in immediate mode.

2. On a lower level, the students may explore screenturtle effects while using single-keystroke commands: F, B, L, R. Versions of this program are available, or can be written, for each implementation of Logo.

Summary:

1. What is the name of the computer language we are learning?

2. What are the commands for turtle turns?

3. What are the commands for turtle moves?

4. What do we type in order to see the screenturtle?

5. How do we erase a letter that we have typed, but don't need?

6. What do we type to erase our drawing and make the turtle go back home?

7. What happens if we forget to put a space between the word FORWARD and the number of steps we want the turtle to take?

Assignments:

1. Make a design using each of the four main Logo commands at least twice. After the design is finished, show how to erase the picture from the screen.

Assignments for More Advanced Students:

1. Experiment to find out how many turtle steps it takes to cross the screen from side to side. How many steps does it take to go from the top to the bottom of the screen?

2. Find out more about the history of Logo. Where did Logo begin? What computer language was it based on? Where did the idea of a turtle come from? Who is Seymour Papert?

TOPIC 27 - E

Logo Lines

Performance Objectives - Students will be able to:

- Get the turtle to draw by commanding it to SHOWTURTLE.
- Improve their concepts of directions: forward, backward, left, and right.
- Command turtle to draw a line by moving FORWARD (FD) and BACK (BK).
- Command turtle to move towards the RIGHT (RT) and the LEFT (LT).
- Press RETURN to make the turtle perform the command.
- Create various shapes using lines.
- Clear the screen of whatever is drawn on it.
- Bring the turtle back to its original position on the screen.

Vocabulary: FORWARD (FD), BACK (BK), RIGHT (RT), LEFT (LT), CLEARSCREEN (CS), HOME.

Movitation: How can we command the turtle to move on the screen?

Aim: To use Logo primitives to move the turtle.

Background Preparation:

FORWARD (FD)	followed by a number, moves the turtle forward
BACK (BK)	followed by a number, moves the turtle backward
RIGHT (RT)	followed by a number , turns the turtle to the right
LEFT (LT)	followed by a number, turns the turtle to the left
CLEARSCREEN (CS)	erases drawing from the screen, and in some implementations of Logo places the turtle in the middle of the screen
HOME	returns the turtle to its home position (center of the screen)

1. Make a turtle out of construction paper. Attach it to a magnet so that it can adhere to and be moved around the chalkboard.

   - Use this turtle model to help the students decide which ways to move the turtle when creating their drawings.

   - Be sure its head is distinct, since it is the pointer in developing the concept of direction.

2. If you are working with very young students with little experience in reading the alphabet, you may want to prepare a diskette, and load the following procedures into their computers:

```
TO F              TO B              TO R              TO L
  FD 20             BK 20             RT 90             LT 90
END               END               END               END


TO C              TO S              TO H              TO U
  CS                ST                HOME              PU
END               END               END               END

TO D              TO E
  PD                PE
END               END
```

. Prepare color-coded letters (F,B,R,L,C,S,H,U,D, and E) to tape over the actual keys so that they will be more apparent.

<u>Development</u>:

1. Rehearse moving the turtle forward, back, right, and left on the chalkboard.

2. At the computer, practice using the commands to move the turtle on the screen like this:

```
Press S  for ST or SHOWTURTLE, and press RETURN
Press F  for FD or FORWARD, and press RETURN
Press B  for BK or BACK, and press RETURN
Press R  for RT or RIGHT; then F
Press L  for LT or LEFT; then F
Press C  for CS or CLEARSCREEN, and press RETURN
```

. Continue practicing these commands in a variety of line arrangements.

. Note: If you are using the regular Logo commands (that is, FD instead of F, etc.), remember that FD, BK, RT, and LT must each have a numerical input following them. Example: FD 30 or RT 90, etc.

Application:

1. Try typing these commands:

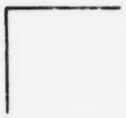
```
FF             FF             F              FRFLFRFL
LF             RFFF           LF             H
BB             LFFFF          LFF            BLBRBLBR
               BBBBBB         LFF            H
                              LFFF           LFRFLFRF
                              LFFF           H
                              LFFF           RBLBRBLB
                                             H
                                             FLFRFLFR
                                             H
                                             LBRBLBRB
                                             H
                                             RFLFRFLF
                                             H
                                             BRBLBRBL
```

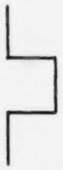
Summary:

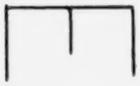
1. How is the turtle made to move?

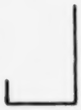
Assignment:

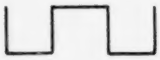
1. Use the commands in this lesson to make these figures


A


B


C


D


E

TOPIC 27 - M

Logo Lines

Performance Objectives - Student will be able to:

- Command turtle to draw a straight line.
- Command turtle to turn towards the right.
- Command turtle to turn towards the left.
- Create various figures using lines.

Vocabulary: FORWARD (FD), BACK (BK), RIGHT (RT), LEFT (LT), HOME, RETURN, CLEARSCREEN (CS), PENERASE (PE), REPEAT

Motivation:

- How can we command the turtle to draw straight lines that go in various directions?

Aim: To use Logo to draw lines in any direction.

Background Preparation:

FORWARD (FD)	followed by a number, moves the turtle forward
BACK (BK)	followed by a number, moves the turtle backward
RIGHT (RT)	followed by a number , turns the turtle to the right
LEFT (LT)	followed by a number, turns the turtle to the left
CLEARSCREEN (CS)	erases drawing from the screen, and in some implementations of Logo places the turtle in the middle of the screen
HOME	returns the turtle to its home position (center of the screen)

Development:

1. Use the FORWARD command with a numerical input, for example: FORWARD 40

   - Note: FORWARD may be entered as FD.

   - Press RETURN to make the turtle perform the command.

2. CLEARSCREEN (CS) and command the turtle to make a shorter line: FD 20

3. CLEARSCREEN and continue to send the turtle forward until it reaches the top of the screen.

   - How far is that?

. Continue sending the turtle forward. It will go off the top edge of the monitor's screen. Continue commanding the turtle FORWARD until it reappears at the bottom of the screen.

- How far is it from the bottom of the screen to the top?

4. CS and send turtle to the right edge of the screen, beginning with the command: RT 90

. Continue sending turtle FORWARD until it reaches the edge.

- How far is this?

. Continue sending turtle FORWARD until it reappears on the left side of the screen.

- Find out the distance from the left side edge of the screen to the right side edge of the screen.

5. Clear the screen and send the turtle FORWARD and BACK along the same track. For example: FD 50 BK 90 or FD 60 BK 30

. In order to turn the turtle to the right or to the left, choose an angle for the turn you want and proceed as follows, for example:

```
RT 45 FD 60 LT 135 FD 100 RT 60 FD 40, etc.
```

. Try this:

```
FD 30
RT 120
FD 30
LT 120
```

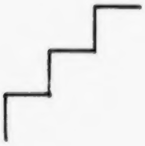
. Repeat those commands. Now try repeating them many times by typing this:

```
REPEAT 9[FD 30 RT 120 FD 30 LT 120]
```

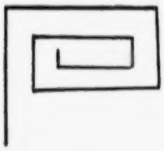
6. CLEARSCREEN.

. Turn the turtle by a small angle, for example RT 5. Send turtle FD or BK with a large input number, for example FD 6000.

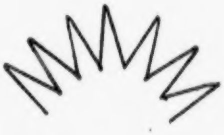
7. Continue using commands: FD, BK, RT,and LT to create these figures:



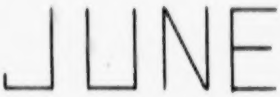
A



B



C



D

- **Note: To see the graphic on a full screen without the printed commands showing, press FS (Atari) or CTRL-F (Apple). SS (Atari) or CTRL-S lets you see the split screen again.**

Applications:

1. Try this:

   - Repeat these commands 72 times:

```
FD 60   BK 60   RT 5
```

   Or type this:

```
REPEAT 72 [FD 60 BK 60 RT 5]
```

Summary:

1. Which Logo commands are used for drawing straight lines?

Assignments:

1. Continue increasing the inputs for FD and BK to very large numbers, and see what happens.

TOPIC 27 - A

Logo Lines

Performance Objectives - Student will be able to:

- Use Logo to draw straight lines while moving either forward or backward.
- Use Logo to make the turtle turn right or left.
- Draw a variety of line figures.

Vocabulary: FORWARD (FD), BACK (BK), RIGHT (RT), LEFT (LT), HOME, CLEARSCREEN (CS), PENERASE (PE), RETURN, Input, REPEAT

Motivation:

- How can we make a variety of figures using lines?

Aim: To use the REPEAT command with Logo primitives to draw lines and designs.

Development:

FORWARD (FD)	turtle moves forward. This command must be followed by a number telling how many steps.
BACK (BK)	turtle moves backward. This command must be followed by a number telling how many steps.
RIGHT (RT)	turtle turns right. This command must be followed by a number telling how many degrees.
LEFT (LT)	turtle turns left. This number must be followed by a number telling how many degrees.
HOME	turtle returns to the middle of the screen, its home position.
CLEARSCREEN (CS)	erases the screen.
PENERASE (PE)	the turtle erases the lines that it moves over.
RETURN	causes the turtle to activate commands.
Input	something that you tell the computer. Certain commands, like FD, BK, RT, and LT, need numbers as part of their input.

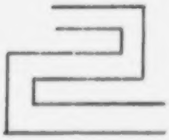
1. SHOWTURTLE (ST)

2. Type these commands:

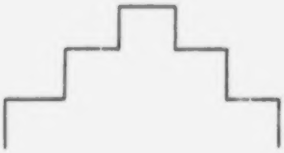
```
FD 20
RT 30
FD 50
BK 80
LT 15
```

3. CLEARSCREEN (CS)

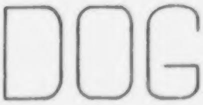
4. Using the above commands, try to make these figures on your computer:



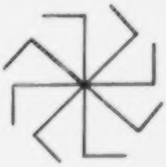
A



B



C



D

5. Try these REPEAT commands

```
REPEAT 8[FD 80 BK 80 RT 45]

REPEAT 72[FD 60 BK 60 RT 5]
```

<u>Summary</u>:

1. When is the REPEAT command used?

<u>Assignments</u>:

1. Try the following REPEAT commands.

   Write down what design each command makes.

   a. REPEAT 10[FD 20 RT 45 BK 20 LT 45]

   This draws ______________________.

   b. REPEAT 30[LT 13 FD 13 LT 13 BK 13]

   This draws ______________________.

   c. REPEAT 100[RT 45 FD 1000]

   This draws ______________________.

d. REPEAT 12[FD 60 LT 20 BK 10 LT 10]

This draws ______________________________.

2. Use REPEAT to make an interesting picture. Write a REPEAT command that looks like this. You add the numbers.

a. REPEAT ___ [FD __ RT __ FD __ LT __ FD __ LT __]

3. Here are 4 REPEAT commands. Guess what kind of design is made by each. Then try each REPEAT command to see the real design.

a. REPEAT 8[BK 30 RT 45 FD 90 RT 90]

GUESS	REAL

b. REPEAT 6[FD 40 RT 60]

GUESS	REAL

c. REPEAT 4[FD 30 RT 30 FD 15 RT 60]

GUESS	REAL

d. REPEAT 8[FD 50 RT 45]

GUESS	REAL

TOPIC 28 - E

PENUP and PENDOWN Commands

Performance Objectives - Students will be able to:

- Lift the turtle's pen so it can move without drawing a line.
- Lower the turtle's pen so it can draw a line as it moves.
- Draw more interesting figures.

Vocabulary: PENUP (PU), PENDOWN (PD), HOME

Motivation:

- How can we move the turle to different places on the monitor screen to draw?

Aim: To use Logo primitives PENUP (PU) and PENDOWN (PD).

Development:

1. Rehearse with the group, using a turtle model on the chalkboard, how to move the turtle in order to make a large equals sign: (=)

2. At the computer, try this:

   - Using program previously prepared for primary grades' use [see Background Preparation for lesson on Logo Lines], type:

```
R  F
U  L  F  L
D  F
```

   - Or, using the regular Logo commands:

```
RT 90
FD 20
PU LT 90
FD 20 LT 90
PD FD 20
```

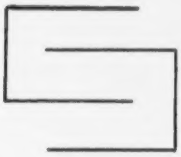
Summary:

1. Which Logo command is used to move the turtle without drawing on the screen?

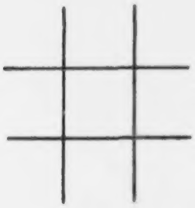
2. Which Logo command is used when you want the turtle to draw again?

__Assignment__:

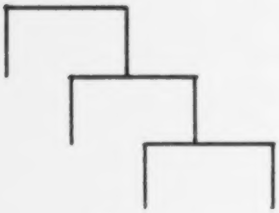
1. Try to draw these figures on your screen:



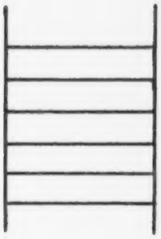
A



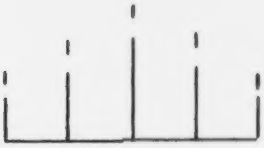
B



C



D



E

28 - A

PENUP and PENDOWN Commands

Performance Objectives - Students will be able to:

- Move the turtle's screen position without leaving tracks.
- Resume drawing when the turtle is in the desired position.
- Draw more complex figures.
- Begin a graphic from a position other than the center of the screen (HOME).

Vocabulary: PENUP (PU), PENDOWN (PD), HOME, PENERASE (PE)

Motivation:

- How can we start drawing in a position other than the middle of the screen (the HOME position)?

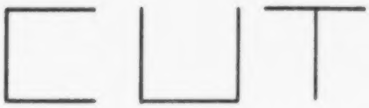
Aim: To use Logo primitives PENUP (PU) and PENDOWN (PD) to draw a complex figure in a position other than the center of the screen.

Development:

1. Review these commands for making lines: FD, BK, RT, LT.

2. To illustrate how the two new commands, PENUP (PU) and PENDOWN (PD), can be used, try these commands to draw the word:

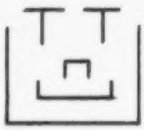
A



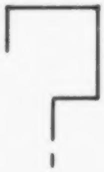
```
LT 90 FD 30
LT 90 FD 50
LT 90 FD 30
PU FD 20 LT 90
PD FD 50 RT 90
PU FD 30 RT 90
PD FD 50 RT 90 FD 30
PU BK 65 RT 90
PD FD 50 LT 90 FD 15 BK 30
```

- Note: If you make a mistake, press PE followed by the commands that will send the turtle over the tracks that need erasing. To resume drawing, press PD before resuming your commands.

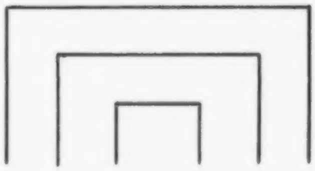
3. Use the PENUP (PU), PENDOWN (PD), and HOME commands to help make these graphics:



B



C



D

Application:

1. Using the new commands, draw the word L I T T L E on the screen. Because of the length of the word, the turtle will need to start the first L some distance left of the HOME position. To do this, begin this way:

   . **SHOWTURTLE (ST)**

   . Left turn at a 90 degree angle : LT 90

. Decide how far over to the left you want to go. Type PU. Then type FD and the number.

   . You may begin the first L by putting the turtle's pen down (PD), and typing FD followed by the number indicating how long you want the bottom line of the L to be. For example:

```
ST
LT 90
PU FD 50
PD FD 20      etc.
```

Summary:

1. How can you make the turtle move to any position on the screen without drawing lines?

Assignment:

1. Continue to create drawings that begin with the turtle in a spot on the screen that is not the HOME position.

TOPIC 29

Hide the Turtle

Performance Objectives - Students will be able to:

- Hide and show the turtle.
- Show a drawing with the turtle removed, so that all of the lines can be clearly seen.

Vocabulary: HIDETURTLE (HT), SHOWTURTLE (ST), HOME, PENUP (PU), PENDOWN (PD)

Motivation:

- How can we remove the turtle to get a better view of our graphic?

Aim: To remove the turtle from the screen after a graphic has been drawn.

Development:

1. The HIDETURTLE command causes the turtle to become invisible. This has two effects:

   - It allows the viewer to see all the lines of the graphic more clearly, without obstruction caused by the turtle's body.

   - The lines are drawn more quickly when the turtle is not visible.

   Press HT - the turtle disappears.
   Press ST - the turtle reappears.

Application:

1. This activity offers the student an opportunity to develop skill at evaluating his/her ability to follow directions for moving the turtle on the screen.

2. Put these procedures into the computer:

```
REPEAT 36[FD 10 RT 10]


REPEAT 36[FD 10 LT 10]
```

3. Type these REPEAT commands.

```
REPEAT 36 [FD 12 LT 10]

REPEAT 36 [FD 10 LT 10]
```

4. Two or more students may take turns in this activity.

   - Press PU, HOME, and HT.
   - One student types in commands that sends the turtle to a place between two of the twelve rings on the monitor screen.
   - Another student then points to the place that he/she judges the turtle to be (using the text of commands that is showing as a guide).
   - The first student presses ST. If the guess is correct, the student wins a point. If the guess is incorrect, no point is won. The next student then sends the turtle and presses HT.

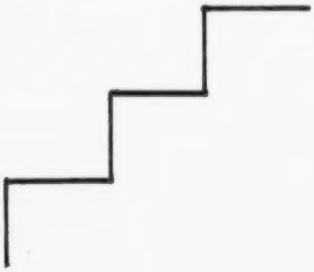
<u>Summary</u>:

1. When our graphic is completed, which command will remove the turtle from the screen?

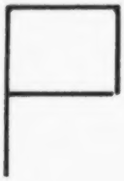
<u>Assignment</u>:

1. Hide the turtle. Try drawing these lines without the turtle's help.

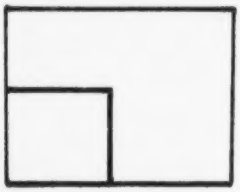
A.



B.



C.



TOPIC 30 - E

The REPEAT Command - Early Experience

Performance Objectives - Students will be able to:

- Explain why the REPEAT command is useful in writing a list of instructions.
- Explain the purpose of each symbol in the repeat command.
- Use the REPEAT command in a list of instructions for making a square, rectangle and triangle.
- Write their own procedures that use the REPEAT command.
- Create their own designs using the REPEAT command.

Vocabulary: REPEAT, pattern

Motivation:

- Repeat several directions, such as: stand, jump, spin, sit, stand, jump, spin, sit.

  - Where did the pattern begin to be repeated?

  - Elicit an easier way to say these directions.

- Do this with patterns that repeat themselves from three to six times.

- Make sure the students can tell you how many times the pattern was repeated and what the pattern sequence was.

Aim: How is the REPEAT command useful in writing a list of instructions?

Development:

1. Discuss with the students the Logo commands used to make a square.

   - Draw a square on the chalkboard.

   - Elicit the definition of a square (an enclosed figure with four equal sides and four equal angles).

   - Elicit commands for making a square. Record these steps putting two commands on each line.

```
FD 50 RT 90
FD 50 RT 90
FD 50 RT 90
FD 50 RT 90
```

   - What patterns can be noticed?

2. Introduce the REPEAT command as a way to simplify writing the program.

   - By telling Logo the number of times to repeat a command or a series of commands, the commands need be written only once.

   - Enclose the commands to be repeated in square brackets.

3. Rewrite the commands on the chalkboard (use the REPEAT command).

```
REPEAT 4 [FD 50 RT 90]
```

   - What is the purpose of each item in the line above?

REPEAT	Tells the turtle to do the commands again.
4	Tells the turtle to repeat the enclosed commands four times.
[ ]	Contains the list of commands.
FD 50	Tells the turtle to move forward 50 steps.
RT 90	Tells the turtle to make a 90 degree right turn.

4. Have the students practice using the REPEAT command.

   - List several other non-REPEAT programs on the board.

   - Create a square that is smaller than those listed.

   - Have the students create a larger square.

5. Create a design using two different sized squares. Brainstorm designs.

   - List the ideas on the chalkboard.

   - Draw one of these designs by having the students give the Logo commands to follow.

   - Have the students create their own designs on the computer using the REPEAT command.

6. What other shapes besides a square have a repeated pattern?

   - List these shapes.

   - Draw a rectangle on the board. Write 50 (turtle steps) along the vertical side and 80 (turtle steps) along the horizontal side.

   - Discuss how rectangles differ from squares.

   - Elicit the definition of a rectangle.

   - Elicit commands for making a rectangle. Record these steps:

```
FD 50 RT 90 FD 80 RT 90
FD 59 RT 90 FD 80 RT 90
```

- Rewrite this procedure using the REPEAT command.

- Create larger and smaller rectangles using the REPEAT command.

7. Create triangles of different sizes using the REPEAT command.

8. Record the programs in Logo logs.

Assignments:

1. Have students create their own designs, using different sized squares, rectangles or triangles. The students should record the programs for their design in their Logo log.

Assignments for More Advanced Students:

1. Create three other shapes using the REPEAT command.

2. On graph paper, create a design that combines squares, triangles, and rectangles of different sizes. Program the computer to make the design.

TOPIC 30 - A

The REPEAT Command

Performance Objectives - Students will be able to:

- Read a REPEAT command and determine each action the turtle would make in the correct sequence.
- Explain what types of triangles can and cannot be made using the REPEAT command.
- Write procedures and create complex designs using the REPEAT command.

Vocabulary: sequence

Motivation:

- How many shapes in the room are made up of a repeated pattern?

- What shapes are they? (squares, triangles, rectangles, stars, polygons, etc.)

- What are some other shapes with repeated patterns?

- If you were to give directions to the turtle to make all these shapes what would be the same in all your instructions? (Repeating, going FD or BK, RT or LT.) What would be different? (The input after REPEAT, FD, BK, RT and LT.)

- Why is the sequence of instructions important in a repeated pattern?

Aim: How can the repeat command be used to make other shapes?

Development:

1. Discuss how the repeat command is useful in condensing a list of instructions.

2. Elicit several REPEAT command programs for making different squares, and list them on the chalkboard.

   - Example:

```
REPEAT 4 [FD 80 RT 90]
REPEAT 4 [FD 50 RT 90]
REPEAT 4 [BK 30 LT 90]
```

   - Which numbers can be changed? (The number after FD and BK)

   - How does the square change when this number changes? (It gets larger or smaller.)

148

- Which numbers cannot be changed? (The numbers after REPEAT and RT and LT) Why? (A square has four sides and angles, and the angles are all right angles.)

3. Predict what the turtle would do if the order of the commands to be repeated in a square program were changed.

   - Write the following two programs on the chalkboard:

   ```
   REPEAT 4 [FD 50 RT 90]     REPEAT 4 [RT 90 FD 50]
   ```

   - Draw the exact path the turtle takes in each instance.

   - What is the difference in the paths? (In the first square the turtle goes forward first and then turns 90 degrees. In the second square the turtle first turns 90 degrees and then goes forward.)

   - Would the squares be exactly the same when they were finished? (No, they will be in different places.)

   - Would the turtle end up in the same position? (Yes) Why? (Because it went in a complete loop.)

4. Create squares of different sizes using the REPEAT command and then reverse the order of commands and write down the results in the log.

5. Have students discover different ways of making procedures for squares that end up looking the same (using BK and LT as well as FD and RT).

6. Predict what the turtle does if the order of commands in a rectangle program were changed.

   - Elicit a program for a rectangle using the REPEAT command.

   - Record this program and then reverse the order of the instructions to be repeated.

   - Write on chalkboard:

   ```
   REPEAT 2 [FD 50 RT 90 FD 80 RT 90]
   REPEAT 2 [FD 80 RT 90 FD 50 RT 90]
   REPEAT 2 [RT 90 FD 50 RT 90 FD 80]
   REPEAT 2 [RT 90 FD 80 RT 90 FD 50]
   ```

   - Draw the four rectangles on paper and then check for accuracy on the computer.

   - Do these rectangles look exactly the same? (No) Why?

   - Create a design with rectangles by writing programs in which the instructions to be repeated are reversed.

7. Hand out WORKSHEET 1.

8. Create triangles of different sizes using the REPEAT command.

   - Draw three kinds of triangles on the board (equilateral, right, and obtuse).

   - Which of these can be written using a REPEAT command? (Equilateral) Why can't the others? (The pattern is not repeated three times.)

   - Elicit commands for making an equilateral triangle.

     ```
     REPEAT 3 [FD 50 RT 120]
     ```

   - Do Projects 1 and 2.

<u>Assignments</u>:

1. Have students create designs on the computer, using different sizes of triangles, squares and rectangles. The designs may be worked out on graph paper first. Procedures should be recorded in their logs.

2. Find pictures of shapes with repeated patterns. Make a scrapbook or collage of these designs. Write a computer program to create one of these designs.

<u>Assignments for More Advanced Students</u>:

1. Write a procedure using REPEAT that makes a pattern and turns the turtle and makes the pattern again and again.

2. Remind students that there are 360 degrees in a circle. Have students solve the following problems and then use the information from these problems to help them create circles using the REPEAT command.

   - How many degrees will the turtle turn in each case?

     ```
     REPEAT 360 [FD 1 RT 1]
     REPEAT 40 [FD 12 RT 90]
     REPEAT 45 [FD 50 LT 8]
     ```

   Create different sized circles using the REPEAT command. Note how much time each takes for the turtle to make. See if the students can come up with any generalizations about the circles.

WORKSHEET 1

1. On this sheet are Logo instructions for the turtle to follow. You are going to see if you can predict exactly what the turtle would do given each set of commands. Under each set of commands draw the design that you think the turtle will draw. After you are finished, check your accuracy on the computer.

```
REPEAT 2 [FD 50 RT 90]           REPEAT 2 [RT 90 FD 50]


REPEAT 4 [LT 60 BK 30]           REPEAT 4 [BK 50 RT 45]


REPEAT 5 [FD 50 RT 90]           REPEAT 5 [FD 50 RT 45]


REPEAT 3 [FD 30 RT 120]          REPEAT 3 [FD 30 RT 60]


REPEAT 3 [FD 40 RT 45 BK 40]     REPEAT 6 [BK 20 LT 30]


REPEAT 5 [FD 50 RT 144]          REPEAT 6 [RT 90 LT 90]
```

TOPIC 31

A Logo Procedure

<u>**Performance Objectives**</u>: Students will be able to:

- Define a procedure
- Name the parts of a procedure
- Write a procedure

<u>**Vocabulary**</u>: Procedure

<u>**Motivation**</u>:

- Did you know that you can teach the turtle many new words?

<u>**Aim**</u>: To introduce the concept and parts of a procedure.

<u>**Development**</u>:

1. Ask the students what teaches them how to make a cake (cookies, jello, etc.)? (A recipe.)
2. Elicit from the students the parts of a recipe (title, ingredients and directions.)
3. Tell them that they can teach the turtle how to make new designs and pictures by writing Logo recipes (Procedures).
4. Here's how to do it:
    a. Start by deciding what name to give your procedure. (It's a good idea to give the procedure a name that's easy to remember. If your procedure draws a square, call it a square.)
    *Always remember to key in the word TO before you name a procedure.

    b. Next think of the command(s) that will draw the name.

    c. Finally, add the word on a separate line after your commands.

    - Example:

```
TO SQUARE
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
END
```

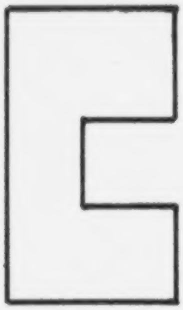
5. Tell the students they can tell the turtle to draw the procedure by typing it's name (i.e. Square) and pressing RETURN/ENTER.

6. Repeat steps 4 and 5 using different procedures.

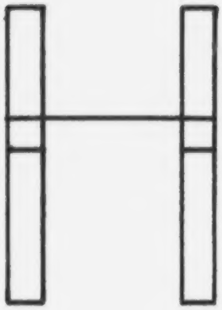
Assignment:

1. Students work in pairs. One walks the turtle to form shapes while the other writes down the commands the turtle will need to draw the shapes.

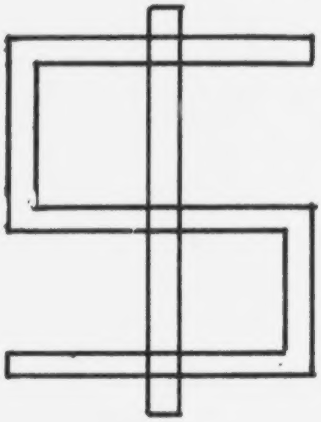
A.



B.



C.



2. Teach the turtle the procedures for the above. Call your procedures to see that they work.

Summary:

A procedure is a command(s) that teaches the turtle to do something.

TOPIC 32

Procedures Inside of Procedures

Performance Objectives - Students will be able to:

. Use a procedure inside a procedure

Vocabulary: Superprocedure

Motivation: How can we make a procedure repeat itself?

Aim: To write a superprocedure.

Development

1. Review the definition and parts of a procedure.

2. Write a procedure for a Square:

```
TO SQUARE
REPEAT 4 [FD 40 RT 90]
END
```

3. Now tell the turtle to draw the Square.

4. Do it again. (Remember to clear the screen.) Call Square again.

5. Now don't clear the screen first. Instead tell the turtle to move forward 40 steps (FD 40) and then call Square.

6. Incorporate the procedure SQUARE with commands to draw the design below. Write a superprocedure for SQUARES.

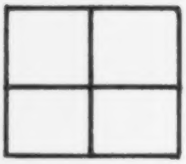
```
TO SQUARE
SQUARE
FD 40
SQUARE
LT 90
SQUARE
END
```

Summary:

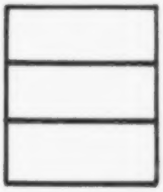
A superprocedure uses one procedure inside another procedure.

Assignments:

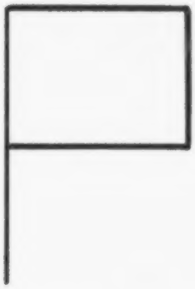
1. Write a superprocedure for SQUARE4.



2. Write a superprocedure for BigWindow

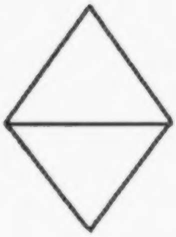


3. Write a superprocedure procedure for FLAG.

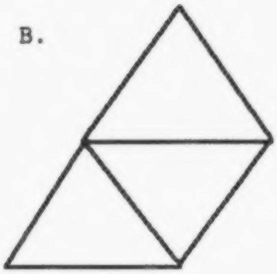


4. Write a procedure for a triangle. Then write a superprocedure for the designs below.

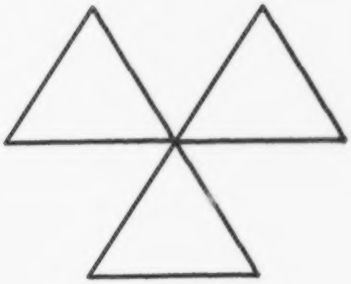
A.



B.



C.



TOPIC 33 - E

Editing in Logo

(For Apple and Terrapin Logo)

Performance Objectives - The students will be able to:

- Explain the difference between immediate (direct) and edit modes and know when to use them.
- Get into and out of each mode.
- Write procedures in the edit mode.

Background:

- Terrapin Apple Logo has two modes: immediate and edit.
  - In immediate mode, the computer immediately does whatever commands are typed.
  - In the edit mode new procedures are defined (written), stored, and edited (changed).
  - Commands are entered in immediate mode, and procedures are entered and edited in edit mode.
- In Apple Logo there are three modes: direct, procedure, and edit.
  - In direct mode, the computer immediately executes whatever commands are entered.
  - In procedure mode, procedures are defined and stored.
  - In edit mode procedures can be defined, stored and edited.
  - Commands may be entered in direct mode, and procedures (sequences of commands that have been given a name) may be entered in procedure mode or edit mode. Procedures are edited only in the edit mode.

Vocabulary: procedure, command, immediate mode, direct mode, edit mode

Motivation:

- How is a computer like a human brain? (It can follow directions, store information (remember things), and do math problems.)
- How is it different? (It can't think; it only does what it is programmed to do.)
- How is it that computers can't make mistakes, but they can do the wrong thing? (The programmer or person who writes directions for the computer, or the one who inputs the information, can make mistakes.)

Aim: How do you teach the turtle to remember a sequence of commands?

Development:

1. Explain what the direct or immediate mode is.

   . Up to now, commands have been typed in and the turtle has been following those commands immediately after the RETURN key is pressed. This mode is called immediate (Terrapin) or direct (Apple). Why does it have this name? (The turtle responds immediately or directly after the return key is pressed.)

2. Elicit the advantages of this mode. (You see your design immediately.)

3. What cannot be done in this mode?

   . What happens if you make a mistake in this mode?

   . Have students experiment with making mistakes in the immediate mode. Have them write FD, BK, RT, or LT without a number after it. Have them misspell a word. Ask them to take note of what happens.

     - What happened? (You got error messages.)

     - Can you fix the errors without retyping everything? (No)

     - Can the turtle remember your design and make it again? (No -- only if you retype it.)

4. Introduce the edit mode.

   . How do the problems of being in immediate or direct mode limit the usefulness of the computer? (The computer can not remember procedures, so they must be retyped to be fixed.)

   . Explain that the computer has a different mode that can store information (remember procedures) and in which the procedures can be fixed (edited). This is called the edit mode.

   . What can you do in edit mode? (You can teach the turtle to make the designs that you want it to make.)

   . Note to teachers using Apple Logo:

     - Explain the procedure mode.

     - You get into procedure mode by writing the word TO before the sequence of commands to be defined.

     - An advantage to this mode is that students can see the graphics immediately after writing their procedures without going back to direct mode. They do this by writing the word END and writing the name of the program after it is defined.

  - However, since they cannot edit in this mode, it can be frustrating. All of these functions plus editing can be done in the edit mode.

4. How do you get into and out of edit mode?

   . In Terrapin Logo you type TO before the procedures to be defined.

   . In Apple Logo you type EDIT or ED and put a " before the procedure to be defined.

5. Elicit the sequence of commands to make a box.

```
FD 59 RT 90
FD 50 RT 90
FD 50 RT 90
FD 50 RT 90
```

   . Explain that to get into edit mode the procedure must be given a name.

     - What is the advantage of giving the procedure a name that looks as much like the design as possible. (It is easier to remember).

     - Tell students that it is best to keep the name as short as possible and that abbreviations may be used. You can not leave any spaces in the name because the computer will only read the first word.

   . Explain to students that they may make more than one shape of the same kind but each time they must save it under a different name or they will replace the original shape with the last one defined. (e.g. BOX1 and BOX2)

   . What would be a good name for the procedures written on the board? (SQUARE, SQ or BOX)

5. Have students do Worksheet 1 on choosing an appropriate name for a procedure.

6. How to get back into immediate or direct mode:

   . Explain to students that you can not see the design you have programmed in edit mode. You must be in immediate or direct mode to see it.

   . To get back to immediate or direct mode, hold down the CTRL key and the letter C at the same time. This tells the computer that the program is complete (C) and it should be added to the dictionary in the memory of the computer.

7. Have students do PROJECTS 1 and 2.

WORKSHEET 1

What would you write before each set of procedures if you wanted to get into edit mode? Remember you must name the procedures. What name would be best for each of the following procedures? You may try each one in immediate mode, if necessary, to help you figure out the shape the procedure makes.

```
FD 100 RT 90   (TO BOX) (TERRAPIN)
FD 100 RT 90   (ED "BOX) (APPLE)
FD 100 RT 90
FD 100 RT 90


FD 100 RT 120    (TO TRI) (TERRAPIN)
FD 100 RT 120    (ED "TRI) (APPLE)
FD 100 RT 120

FD 100 RT 90     (TO REC) (TERRAPIN)
FD 50 RT 90      (ED "REC) (APPLE)
FD 100 RT 90
FD 50 RT 90
```

PROJECT 1

Write the 3 procedures on Worksheet 1 in the edit mode.

Make sure that you first write TO (or ED ") and the name of the procedure and then press the return key.

Write the procedures one at a time.

Press the return key after each line.

Give the Logo commands to make one of the shapes that are given on the worksheet. When you are done, show you are finished by typing END.

Press down the CTRL key and type the letter C to get back to immediate (or direct) mode.

Now that the turtle can make the shape, type the name of the shape to see it.

Do the same thing with the second and third shape.

PROJECT 2

Use the 3 shapes you have taught the turtle in PROJECT 1 to make a picture. Some suggestions are...a house, a tree, a flag, or a design of your own. In order to get the shape on the screen, all you have to do is write the name of it in immediate mode and press the return key.

TOPIC 33 - A

Editing in Logo

(For Apple and Terrapin Logo)

Performance Objectives - The students will be able to:

- Use editing controls to move the cursor around.
- Write procedures in the EDIT mode.
- Use editing controls to make simple changes in programs.
- Save procedures.
- Call up procedures.

Vocabulary: procedure, command, immediate mode, direct mode, edit mode

Motivation:

- Write the following statement on the board and ask the students what it means and why this statement was probably made?

A computer does not do what you want it to do,
it does what you tell it to do!

Aim: How do you teach the turtle to remember a sequence of commands? What are the editing commands and how do you use them? How do you save programs?

Development:

1. Review what the direct or immediate mode is.

   - In immediate mode (Terrapin) or direct mode (Apple), the turtle responds immediately or directly to commands after the return key is pressed.

   - Elicit from the students the advantages of this mode. (You see your design immediately.)

2. Review what the edit mode is.

   - Elicit from students that the computer has a different mode that can store information (remember procedures) and in which the procedures can be fixed (edited) called the edit mode.

   - What can you do in Edit Mode? (You can teach the turtle to make the designs that you want it to make.)

3. Review process for getting into and out of edit mode.

   - In Terrapin Logo you Type TO before the procedures to be defined.
   - In Apple Logo You type EDIT or ED and put a " before the procedure to be defined.
   - What must you do to get into edit mode? (You must give your procedure a name and type TO or ED " before the name.)
   - Why do you need to get back into immediate (or direct) mode when you are finished writing your procedure? (You cannot see the design in edit mode. You must be in immediate or direct mode to see it.)
   - How do you get back into immediate mode? (T get back to immediate or direct mode, hold down the CTRL key and the letter C at the same time. This tells the computer that your program is complete (C) and it is added to the dictionary in the computer's memory.)

4. Give out the EDITOR SHEET. Go over the commands on it with the group.

5. Have students do WORKSHEET 1.

6. Have students do PROJECTS 1-5.

7. Have students save their procedures on their own disks by typing SAVE " procedure name. All procedures that they have defined since turning on the computer will be saved.

8. How do we get back a program that we saved before? Ask students why they may want to get back a program they saved before. (They may want to work on it again, or use parts of it for a new program.)

   - Have students turn off their machines after they have saved their program to disk. Tell them to do the following to load a program they have saved:

     - Reboot disk (by first booting language disk and then booting their initialized disk).
     - Type the word catalog to see the name of all the procedures saved.
     - Type READ " name of procedure saved (Terrapin) or LOAD "name of procedure saved (Apple). Explain to the students that the computer will call up all of the procedures they saved under that name.
     - Retype the name of the procedure you want to see and press the return key.

9. Have students do Projects 6 and 7.

WORKSHEET 1

TEACH THE TURTLE THE PROCEDURE LINE.

- First enter the edit mode and tell the turtle the name of the procedure you want it to remember by typing TO LINE or ED " LINE.
- Define the commands in the procedure by giving the Logo commands to make a line.
- When you are done show you are finished by typing END.
- Type CTRL C to get back to immediate mode.
- Now that the turtle knows LINE type the word LINE and see what the turtle draws.

==

PROJECTS

1. Teach the turtle the procedure LONG.LINE. Remember, if you want to teach the turtle a procedure that has more than one word in it's name, connect the words with a dot, like LONG.LINE. Otherwise, only the first word will be read.
2. Teach the turtle to make the letter V. Try it in immediate mode first and then in edit mode. When you want to try it in edit mode, begin by writing TO VEE (Terrapin) or ED "VEE (Apple).
3. Teach the turtle to make a box.
4. Teach the turtle to make a smaller box. Remember to give it a different name than the first box.
5. Teach the turtle to make a triangle and a rectangle.
6. See what new words are in the turtle's dictionary by typing the word POTS. This stands for Print Out Titles. Write these words down.
7. Save all of your procedures by typing SAVE "procedurename. Give your procedures one name like DESIGNS. All of the procedures you have written since you turned on the computer will be saved.
8. Now turn off the computer.

9. Reload the language disk and your own file disk. Call up your procedures by typing READ "DESIGNS (Terrapin) or LOAD "DESIGNS (Apple). Use your old procedures to make a new design.

. Tell students to use the EDITOR SHEET if they need to correct mistakes)

EDITOR SHEET

This chart shows ten useful editing commands for making changes in procedures. (See the attached sheet for illustrations where relevant.)

1.	ESC	(Terrapin and Apple)	Moves cursor left and erases the character to the left.
2.	CTRL A	(Terrapin and Apple)	Moves curser right without erasing.
3.	CTRL B	(Terrapin and Apple)	Moves cursor left or back without erasing. (BACKWARDS)
4.	CTRL D	(Terrapin and Apple)	Deletes the character under the cursor. (DELETE)
5.	CTRL K	(Terrapin and Apple)	Deletes the character from cursor position to the end of the line. (KILL)
6.	CTRL N	(Terrapin and Apple)	Cursor goes to the next line without erasing. (NEXT)
7.	CTRL P	(Terrapin and Apple)	Cursor goes to the previous line without erasing. (PREVIOUS)
8.	CTRL O	(Terrapin and Apple)	Opens a space to insert a line. (OPEN)
9.	CTRL C	(Terrapin and Apple)	Leaves edit mode, returns to immediate mode, and adds the procedure to the dictionary. (COMPLETE)
10.	CTRL G	(Terrapin and Apple)	Leaves the Edit mode without defining the procedure. (GONE)

TOPIC 34-E

Squares

Performance Objectives - Students will be able to:

- Create a square with the Logo turtle.
- Use the squares to create designs.

Vocabulary:

Motivation:

- The following can be done as a brainstorming activity with the whole class or by dividing the class into two teams and seeing which team comes up with the most responses.
- Name as many objects in the room that you can that are square shaped. Which objects are made up of many squares? (Elicit any cube shaped objects) Which objects have squares hidden in them?

Aim: What is a square? What designs can be made out of squares? How can the turtle be instructed to make squares and designs out of squares?

Development:

1. Elicit from the students the properties of a square.
   - Hand out five squares of different sizes and colors.
   - Ask students to explain in what way the squares are different.
   - Ask students to explain in what ways the squares are alike. (closed figures, straight sides, equal length on all sides, same shape corners)
   - Elicit the definition of a square from these observations. (an enclosed 4-sided figure with equal sides and angles)
2. Elicit turtle commands for making a square.
   - Play Turtle Robot Game:
     - Tell students that you are the Turtle Robot. You will move when you are given the correct instructions to do so. The students are to try to direct you to make a square. You are beginning in the HOME position.
     - Move only if the student tells you to go FD or BK, RT or LT with a number after it.
     - If they are confused, tell them you will begin again in the HOME position. Otherwise, try not to say anything during this process.

- Once they have successfully given you the directions, draw a square on the board and write down the instructions one at a time.

```
FD 50 RT 90
FD 50 RT 90
FD 50 RT 90
FD 50 RT 90
```

3. Ask which number can be changed and which can't. Why is this true?

4. What else could be changed? (The turtle could go BK instead of FD, LT instead of RT.)

5. Have students work in pairs making different sized squares with the FD, BK, RT, and LT commands. Their programs should be recorded in their computer logs.

6. Have students make designs out of squares that connect. (Insert diagram)

   . Have students work in pairs. Redistribute the colored squares.

   . Ask the students to try to arrange them in nine different ways.

   . One student from each pair can manipulate the squares while the other one notes and records the patterns.

   . Ask the students to select their favorite design, and draw that design on paper numbering the squares in a logical order. What would the turtle have to do after making the first square before making the second square? (move FD or BK and/or move RT or LT)

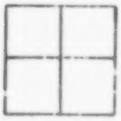
7. Have the students create a tennis court on the computer. They should record their program in their logs.

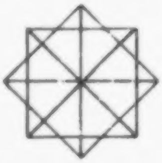
8. Have students make a square, turn RT 45 degrees and make another square. Have them experiment making designs with squares in which they change the angle degrees.

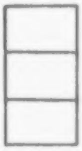
<u>Assignments</u>: 1. Draw the following items on the computer.

. A window . A pinwheel

. A ladder . A sidewalk







<u>Assignment for More Advanced Students</u>:

1. Create a new design with the colored squares. Program the computer to make that design.

TOPIC 34-M

Squares

Performance Objectives - Students will be able to:

- Use Logo to create a square.
- Rotate squares to create designs.
- Use the box shape in new procedures to create new designs.

Vocabulary: rotation, axis

Motivation:

- What designs can be made out of squares? Have students create a scene on drawing paper using only squares.

Aim: How can the box shape be used to create a star? How can the box shape be used to make complex designs?

Development:

1. Create a box.

   - Draw a square on the board.

   - Review procedures for making a square.

```
TO BOX
  FD 50 RT 90
  FD 50 RT 90
  FD 50 RT 90
  FD 50 RT 90
END
```

   - Rewrite these procedures using the REPEAT command.

```
TO BOX
  REPEAT 4 [ FD 50 RT 90]
END
```

2. What type of design would we get if we rotated a square around an axis?

   - Ask students to make a box shape on the computer. Have them alternate rotating the turtle 45 degrees (by typing RT 45) and making another box, until there are boxes going around in a complete circle.

   - Have students count the boxes and RT turns when this activity is completed. (They can do this easily by pressing CONTROL T to see the full text screen.)

3. Elicit a one-line procedure to make this design.

```
TO STAR
  REPEAT 8[BOX  RT 45]
END
```

. Have students answer the following questions:

- Why did we REPEAT 8? (to draw 8 boxes and to make 8 turns of 45 degrees)
- How many degrees did the turtle turn all together? (8 * 45 or 360 degrees)
- How many degrees are there in a circle? (360)
- How would we make the design larger? (change the number after FD)
- What would we need to do if we wanted a design with more boxes? (Increase the number after repeat and decrease the number after RT)
- What do we need to do if we want a design that goes in a complete cirle? (The number after REPEAT multiplied by the number after RT must be 360.)

4. Have students use this information to make new rotating square designs. (see Projects 1 and 2.)

5. Have students draw four boxes that meet.

6. Draw three boxes that are diagonal.

7. Students should do WORKSHEET 1.

8. How can we draw two boxes that don't touch?

   . Review PU (PEN UP) and PD (PEN DOWN) commands.

   . Have students draw two boxes that don't touch.

9. Students should do PROJECT 3.

<u>Assignment for More Advanced Students</u>:

1. Choose one of the following ideas and create the picture on the computer.

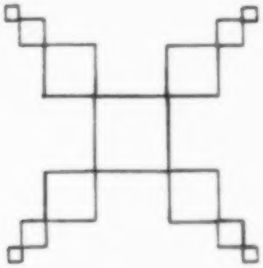
   . Make your own robot out of squares.

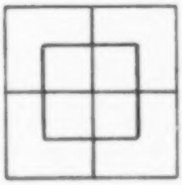
   . Make an apartment building with a sidewalk in front of it out of squares.

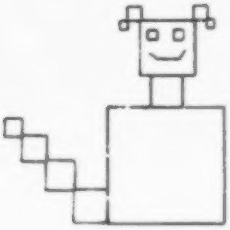
   . Make a new type of vehicle out of squares.

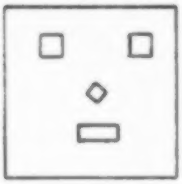
WORKSHEET 1

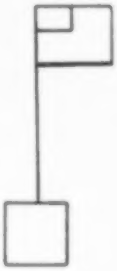
Here are some ideas for using squares. See how many of these you can make on the computer.

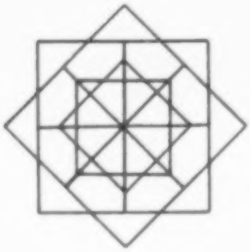












NOW MAKE SOME NEW DESIGNS OF YOUR OWN.

PROJECTS:

1. Write a program to make a rotating square design. Increase the size of the design. Decrease the size. Have the design go around in the opposite direction.
2. Write a rotating square program that has more squares. Write one that has fewer squares. Call up one procedure in the next, just as TO STAR uses BOX.
3. Write a program that makes two or three stars that don't touch.

TOPIC 35 - E

Triangles

Performance Objectives - Students will be able to:

- Use Logo to create a triangle.
- Use a triangle to create designs.

Vocabulary:

Motivation:

- What objects in the room are shaped like a triangle? Are they all exactly the same? How do they differ? How are they the same? Are there any objects that you can think of that are made up of more than one triangle?

Aim: What is a triangle? How do triangles differ from each other? How are they alike? How can the Logo turtle be instructed to make triangles and to make designs out of triangles.

Development:

1. Elicit the definition of a triangle.

   - Hand out four different triangles to each student (one equilateral, one right, one acute and one obtuse).

   - Ask students to observe similarities between them and list these similarities. (All are enclosed figures with three sides and three angles.)

   - Ask students to observe differences between them. (They do not have the same size angles.)

   - Explain the names of the different triangles and elicit why they have that name. (Equilateral triangles have equal length sides and equal angles, right triangles have a right angle, obtuse triangles have an angle larger than 90 degrees, acute triangles have three angles that are not equal but that are smaller than 90 degrees.)

2. On paper, have students draw triangles of the four different shapes.

   - Have students draw a square and divide it diagonally to make two triangles.

   - Which type of triangle is made by dividing a square in half?

   - Since the students have already learned that a square has four 90-degree angles, see if they can tell you how many degrees there are in a right triangle.

   - Do obtuse, acute and equilateral triangles also have 180 degrees?

3. Elicit procedures in Logo for making an equilateral triangle.

   - Draw the following equilateral triangle on the board.

   - How many degrees are there in each angle? (60)

   - How many degrees are there in a straight line? (180)

   - Have students tell you how to draw an equilateral triangle. Have them explain one procedure at a time to make the equilateral triangle on the board. Go over the triangle with colored chalk as they tell you the correct procedure. Ask students what number of degrees the turtle has to turn from going straight (or 180 degrees) in order to have a triangle with three 60-degree angles.

   - Elicit the following procedures from the students:

     ```
     FD 50
     RT 120
     FD 50
     RT 120
     FD 50
     RT 120
     ```

   - How could we change the procedures and still make an equilateral triangle? (We could change the size, go BK instead of FD and go LT instead of RT)

   - Have students make the turtle draw larger and smaller equilateral triangles, using FD or BK, RT or LT.

4. Have students make right triangles by making a square and dividing it in half.

5. Have students make obtuse triangles of various lengths and with different sized angles.

6. Have students make acute triangles of different lengths and sizes.

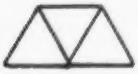
7. Have students do WORKSHEET 1.

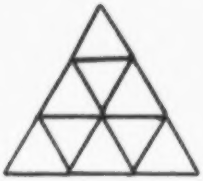
8. Allow students time to experiment making their own designs using triangles, rectangles and squares. Have them record their designs in their Logo logs. Remind them that they can use PU (PENUP) and PD (PENDOWN) commands to hide turtle tracks. Encourage them to share their drawings and designs with the class and explain how they made them (Logo Show and Tell).

9. Have students do Projects 1 and 2 and WORKSHEET 2.

Worksheet 1

CREATE THE FOLLOWING MULTI TRIANGLE DESIGNS ON THE COMPUTER





Worksheet 2

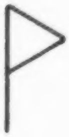
CREATE THE FOLLOWING DESIGNS ON THE COMPUTER, USING LINES, BOXES, RECTANGLES AND TRIANGLES

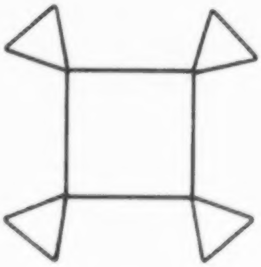
A FLAG

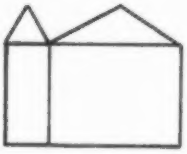
A FAN

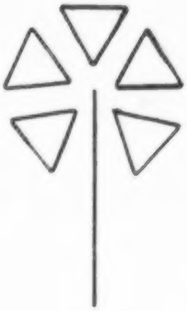
A BARN

A FLOWER





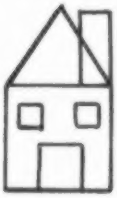


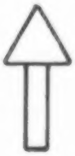


PROJECTS:

1. Draw a house with a chimney, using a square, triangle, and rectangle.

2. Draw a tree using a rectangle and a triangle.





TOPIC 35 - A

Triangles

Performance Objectives - The students will be able to:

- Use Logo to create a triangle.
- Rotate triangles to make designs.
- Use the triangle in new procedures.

Motivation:

- How are triangles different from squares? (3 sides and angles, not 4; angles total 180 not 360 degrees).

- In what ways are they similiar? (Both have straight sides, angles,and are enclosed figures, etc.).

- What are some things that have a triangular shape?

- Have students draw a picture entirely out of triangles.

Aim: How can the triangle be used to make complex pictures and designs in Logo?

Development:

1. Review the four different types of triangles.

   - Elicit their similarities and differences and list these characteristics under appropriate headings on the chalkboard.

     - Similarities: all triangles have three sides and angles, all have angles that add up to 360 degrees, and all are enclosed.

     - Differences: an equilateral triangle has three equal sides and angles, an acute triangle has three angles less than 90 degrees, an obtuse triangle has an angle greater than 90 degrees, and a right triangle has one right angle.

2. Elicit a one-line procedure that could be used to make a triangle.

   - Which type of triangle could be made using a REPEAT 3 command? (Only with the equilateral triangle do you instruct the turtle to do the same thing three times.)

   - What are the commands for making an equilateral triangle if the length of each side is 80 turtle steps?

```
FD 80 RT 120
FD 80 RT 120        or        REPEAT 3 [ FD 80 RT 120]
FD 80 RT 120
```

3. Have students experiment with making equilateral triangles. Tell them to try having the turtle turn RT 60 degrees instead of 120 degrees. Ask them to try to discover the answer to this question: If an equilateral triangle has three 60-degree angles, why does the turtle have to turn 120 degrees and not 60 degrees to make an equilateral triangle? When they are finished, discuss their observations.

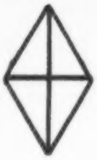
4. Have students create procedures for making the other three types of triangles. (If they have difficulty, review the previous lesson on making triangles, "Triangles Early Experience.")

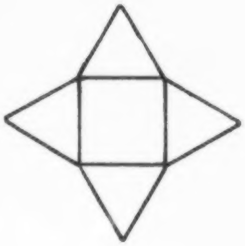
5. Making patterns and designs out of triangles:

   . Diagonally divide a square into four parts to get four triangles. What type of triangles did you get? (RT triangles) How can you tell? (Lines that cross in the center are right angles.)

   . Have students write a procedure for a making rotating triangle design. (If they have difficulty, review procedures for rotating a square and rectangle.) Have them do WORKSHEET 1.

   . Create a triangle from a diamond.
   Experiment with sides of varying lengths and angles.





6. Draw a square inside four triangles.

7. How to use the triangle in more complex procedures:

   . Have students choose projects (see project ideas 1-10).

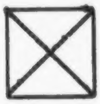
WORKSHEET 1

What can the turtle do after it knows how to make a triangle?

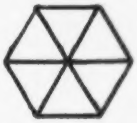
Teach the turtle to make a triangle:

```
TO TRI
   REPEAT 3 [FD 50 RT 120]
END
```

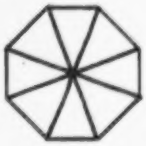
Now have the turtle draw the designs below. Create new procedures that use the command TRI and write them below. When you are finished, test it out on the computer.



PROCEDURE
TO TRI4



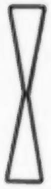
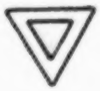
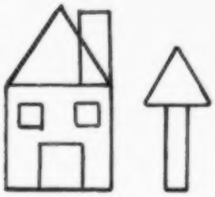
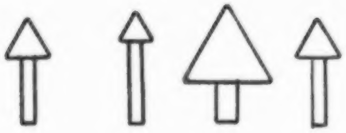
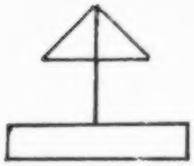
TO TRI6



TO TRI8

PROJECT IDEAS   Choose at least 3

1. Create procedures to make the following pictures on the computer.

- Make 2 triangles that touch.
- Create a procedure that draws one triangle inside another. The triangles should not touch.
- Write a procedure that uses a rectangle, square and a triangle to make a picture. You may work out your design on paper first.
- Draw a house with a tree next to it.
- Draw a forest with trees of various sizes.
- Make a boat out of a rectangle and two triangles.
- Make a flag
- Create your own monster with a square head, triangular eyes, etc. Let your imagination go wild!

2. Create your own scene or design using the shapes you have learned.

Assignment for More Advanced Students:

1. Write a procedure for making a triangle with size as a variable. Call up this procedure to make a butterfly. Make the wings on the butterfly out of triangles that are inside each other. Make antennae and feelers.

TOPIC 36 - E

Rectangles

Performance Objectives - Students will be able to:

- Use Logo to create a rectangle.
- Use the rectangles to create designs.

Vocabulary:

Motivation:

- Look around the room. What objects are made out of rectangles?
- Which are made out of more than one rectangle?
- In what ways are these objects alike? In what ways are they different?

Aim: What is a rectangle? How can the turtle be instructed to make rectangles and designs out of rectangles?

Development:

1. Elicit the definition of a rectangle.
    - Give out a construction paper square and rectangle to each student.
    - Have the students examine them and brainstorm the ways they are alike and different.
    - Elicit the definition of a rectangle from these observations. (An enclosed 4-sided figure, with opposite sides equal and parallel, and all angles right angles.)
2. Elicit from students how to get the turtle to make a rectangle.
    - How did we make a square? (We had the turtle go forward the same amount and make a right turn four times.)
    - What is the difference between a square and a rectangle? (In a rectangle the turtle does not go forward the same amount each time.)

- Draw a rectangle on the board that is twice as wide as it is high. Put the number 50 on the vertical side and the number 100 on the horizontal side. Place an x in the lower right hand corner. Tell the students that this is the home position. Elicit from students the Logo procedures to make a rectangle and to leave the turtle in the home position.

```
FD 50
RT 90
FD 100
RT 90
FD 50
RT 90
FD 100
RT 90
```

- Ask the following questions:

  - What would happen if we changed the number after FD? (We would change the size of the figure.)

  - Would we end up with a rectangle? (It depends on whether we make the opposite sides the same length.)

  - What would happen if we changed the number after RT? (We would no longer have a rectangle, because the angle would no longer be 90 degrees.)

- Elicit alternate procedures for making a rectangle. Erase the x that showed the first home position for the turtle. Put the x in each of the other three corners, one at a time, and have students explain the procedures for making the same rectangle.

3. Have students do Projects 1, 2, and 3.

4. Have students make designs out of rectangles that connect.

   - Ask students to draw and cut out three different rectangles.

   - Have students work in pairs arranging the rectangles in different ways so that they are touching.

   - Have students pick out one of these configurations that they would like to make on the computer.

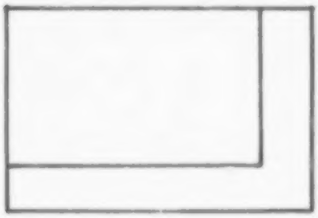
   - Have the students number the rectangles in a logical order.

   - What would the turtle have to do after making the first rectangle, before making the second? (Move FD or BK and/ or RT or LT.)

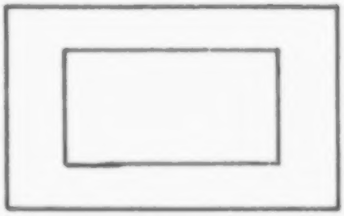
   - Remind students that they can use PU or PENUP and PD or PENDOWN if they do not want to show turtle tracks during a move.

- Have students make the rectangle configuration on graph paper and then on the computer.

5. Draw a rectangle inside a rectangle.



A



B

6. Have students make a rectangle, turn RT 20 degrees and make a new rectangle, etc. Have them experiment with making designs with rectangles, where they rotate the rectangle around an axis.

7. Have students do Worksheets 1 & 2.

8. Allow the students to experiment making their own designs using rectangles and squares. Have them record their designs in their Logo logs.

Assignments:

1. Make the same rectangle two times. One time have the turtle go to the right. The other time have the turtle go to the left.

2. Make at least two different sized rectangles. Use different size turtle tracks.

3. See what happens when you change the angles. Do you still have a rectangle? Why or why not?

Assignments for More Advanced Students:

1. Create one of the following scenes using rectangles and squares:

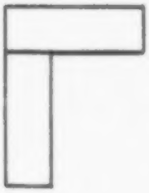
   - An apartment building with a door and windows.

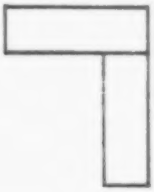
   - A train with several cars and windows.

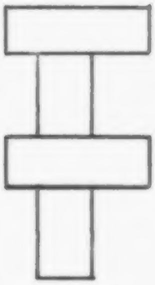
   - A cube.

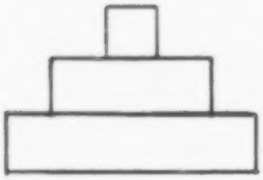
Worksheet 1

Try to draw the following shapes on the computer.



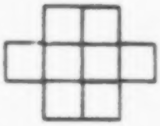


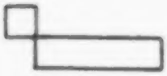


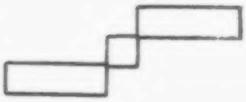


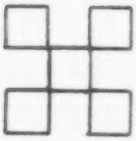
Worksheet 2

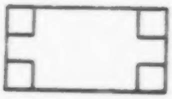
Make the following shapes.











TOPIC 36 - A

Rectangles

Performance Objectives - Students will be able to:

- . Use Logo to create a rectangle.
- . Rotate rectangles to make designs.
- . Use the rectangle in new procedures to create new designs.

Vocabulary:

Motivation:

- . What designs can be made out of rectangles? Have students create pictures on drawing paper using only rectangles.

Aim: How can rectangles be used to make a complex design?

Development:

1. Create a rectangle. Draw a rectangle on the board. Review procedures for making a rectangle. Have students give the different commands required if the turtle started in a different position. Rewrite these procedures using the REPEAT command. Review commands for making larger and smaller rectangles.

2. Elicit the commands the students used for rotating a square around its corner (REPEAT 8 [SQUARE RT 45])

   - . What other commands could have been given to the turtle to make it rotate the square in a complete circle? (REPEAT 10 [SQUARE RT 36], REPEAT 9 [SQUARE RT 40] etc.)

   - . How do we make sure that the boxes will go around in a complete circle? (The number after REPEAT * the number after RT OR LT) must be 360)

   - . Have students do Worksheet 1.

3. Have students rotate different size rectangles around an axis. Have them rotate their design first to the right and then to the left.

4. Have students create a tree or flower by using a rectangle for the stem and rotating a small rectangle or square around the top for the flower.

5. Have students create a fan by rotating the turtle in a semicircle.

6. From a rectangle, have students create two rectangles that touch.

```
TO TWO.REC
  REPEAT 2 [REC RT 90]
END
```

. Have students create a procedure to draw a rectangle inside a larger rectangle.

7. Have students create a procedure to draw the rectangles inside each other that don't touch. Remind them to use PU (PEN UP) and PD (PEN DOWN) commands.

   . Have them spin the new design around one of its corners.

8. Have students write a procedure that uses both a square and a rectangle.

9. Have students do Projects 1-2-3.

Assignments for More Advanced Students:

1. Make a Logo quilt. Write a procedure to divide the monitor into four parts. Write procedures for making rectangular designs by rotating different sized rectangles around an axis. Write a procedure for putting one of these designs in each part of the monitor.

WORKSHEET 1

Which of the following procedures would make the rectangle revolve in a complete circle?

REPEAT 36 [REC LT 10] (yes)

REPEAT 10 [REC RT 36] (yes)

REPEAT 40 [REC RT 8] (no)

REPEAT 40 [REC LT 9] (yes)

REPEAT 20 [REC RT 12] (no)

REPEAT 18 [REC LT 20] (yes)

Projects:

1. Write procedures to make a beautiful fan. Put designs on each of the fan's slats.
2. Create a design using at least 3 different sized rectangles. Some ideas are: a robot, an apartment, a house, a fence, a train, a bus, etc.
3. Combine your shapes. Write a procedure for putting a square inside a rectangle. Put a triangle inside the square!!

TOPIC 38 - E

Colors

Performance Objectives - Students will be able to:

- Change the screen's background color.
- Change the turtle's pen color.

Vocabulary: SETBG, SETPC

Motivation:

- Note: This topic was prepared for Apple Logo.
- Demonstrate the available background colors using Handout #1.
- Demonstrate the available pen colors using Handout #2.
- Copy PIE (Handout #3).
- How can PIE be changed so that the pie (circle) is made larger or smaller?
  - Use the Editor to change all of the 56's in PIE to a different number.

Aim: How can we use the color commands to change procedures?

Development:

1. Introduce SETBG and SETPC as new commands.
2. Using the original PIE procedure:
   - What happens if only the FD commands, and not the BK commands are changed?
   - What happens if LT 1 is used instead of RT 1?
   - What happens if RT 1 is changed to a larger angle?

Applications:

1. Write a new procedure which makes a pie with five colors instead of six.
2. Make a pie with four colors.
3. Add SETPC or SETBG to any previously defined procedure.

Summary:

1. How can SETBG and SETPC be used to change procedures?

Assignment:

1. Use PIE or any one of its variations as a subprocedure in an original program. For example, a program which makes a car can be changed so that it has the cut up circle (PIE) instead of plain circles as wheels. (See Handout #4.)

Handout #1:

```
SETBG 5 (blue)
SETBG 4 (orange)
SETBG 3 (violet)
SETBG 2 (green)
SETBG 1 (white)
SETBG 0 (black)
```

Handout #2:

```
SETPC 5 (blue)
SETPC 4 (orange)
SETPC 3 (violet)
SETPC 2 (green)
SETPC 1 (white)
SETPC 0 (black)
```

Handout #3:

```
TO PIE
  REPEAT 60[SETPC 0 FD 56 BK 56 RT 1]
  REPEAT 60[SETPC 1 FD 56 BK 56 RT 1]
  REPEAT 60[SETPC 2 FD 56 BK 56 RT 1]
  REPEAT 60[SETPC 3 FD 56 BK 56 RT 1]
  REPEAT 60[SETPC 4 FD 56 BK 56 RT 1]
  REPEAT 60[SETPC 5 FD 56 BK 56 RT 1]
END
```

Handout #4
Car with regular wheels:

```
TO CAR
  LT 90 FD 70 LT 90 WHEEL RT 90
  PU FD 35 PD FD 35 RT 90 FD 25
  RT 90 FD 25 LT 45 FD 40 RT 45
  FD 70 RT 45 FD 40 LT 45 FD 50
  RT 90 FD 25 RT 90 FD 25 LT 90
  WHEEL
END
```

```
TO WHEEL
   REPEAT 36[FD 3 RT 10]
END
```

Car with "pin" wheels:

```
TO FANCYCAR
  LT 90 FD 87 PINWHEEL PU FD 18 SETPC 1
  PD FD 35 RT 90 FD 25
  RT 90 FD 25 LT 45 FD 40 RT 45
  FD 70 RT 45 FD 40 LT 45 FD 50
  RT 90 FD 25 RT 90 FD 42
  PINWHEEL
END

 TO PINWHEEL
   REPEAT 72[SETPC 3 FD 17 BK 17 RT 1]
   REPEAT 72[SETPC 4 FD 17 BK 17 RT 1]
   REPEAT 72[SETPC 5 FD 17 BK 17 RT 1]
   REPEAT 72[SETPC 1 FD 17 BK 17 RT 1]
   REPEAT 72[SETPC 2 FD 17 BK 17 RT 1]
 END
```

TOPIC 38- A

Colors

Performance Objective - Students will be able to:

- Write procedures that will change the screen's background color.
- Write procedures that will change the turtle's pen color.

Vocabulary - New: SETBG, SETPC.
Review: Writing procedures, WAIT

Motivation:

- Note: This topic has been prepared for Apple Logo.

- How can FLASH be changed so that the background colors remain on the screen longer?

```
TO FLASH
  SETBG 5
  SETBG 4
  SETBG 3
  SETBG 2
  SETBG 1
  SETBG 0
END
```

- Make FLASH a recursive procedure.

- Use WAIT and an input after each color change.

- Copy COLORS (Handout #1).

Aim: How can we use color changes to enhance procedures?

Development:

1. Review commands which need input. (Commands, such as CS, HT, and HOME, need no input.) Many Logo commands need inputs before any action takes place. Examples: FD 93, WAIT 200

2. Introduce SETBG and SETPC as new commands.

3. Refer to the Motivation and elicit a similar procedure that demonstrates the six pen colors available.

    - In other versions of Logo, where there are more than six colors, suggest a manageable number that will show a fair representation.

```
TO CIRCLE
  SETPC 5 REPEAT 6[FD 8 RT 10]
  SETPC 4 REPEAT 6[FD 8 RT 10]
  SETPC 3 REPEAT 6[FD 8 RT 10]
  SETPC 2 REPEAT 6[FD 8 RT 10]
  SETPC 1 REPEAT 6[FD 8 RT 10]
  SETPC 0 REPEAT 6[FD 8 RT 10]
END
```

. CIRCLE is just one example of a procedure which shows the spectrum of pen colors available. Any similar procedure is an acceptable response.

<u>Applications</u>:

1. Write a new procedure that uses SETBG and SETPC in any of its subprocedures.

2. Add SETBG and SETPC to a previously defined procedure.

<u>Summary</u>:

1. How can SETBG and SETPC be used to enhance procedures?

2. Why does the WAIT command often add variety to the SETBG command?

<u>Assignments</u>:

1. Using FLASH, change the order of the background colors so that the procedure ends with the color orange.

2. Write a program that demonstrates the following:

   . Geometric figures with different colored sides.

   . A program demonstrating "day" and "night" using different background colors.

Handout #1:

```
TO COLORS
  SETBG 0
  SETBG 1
  SETBG 2
  SETBG 3
  SETBG 4
  SETBG 5
END
```

TOPIC 39

"Walking Geometric Shapes" Like a Turtle

Performance Objectives - Students will be able to:

- Follow directions for experiencing geometric shapes through body geometry.
- Exhibit competence in directing another student to walk a geometric shape.
- Recognize the unique qualities of the regular polygons: triangle, square, pentagon and octagon.
- Recognize the relationship between increasing the number of sides in a regular polygon and increasing the resemblance of the polygon to a circle.

Vocabulary - New: triangle, square, pentagon, octagon, circle
Review: commands, trainer, procedure

Motivation:

- What types of activities have we done while "playing turtle?" (Responses may include: walking a maze, turning, moving, reaching a goal, etc.)

Aim: How can we follow or give directions for walking geometric shapes as the turtle?

Development:

1. On the board or chart, display pictures of a triangle, square, pentagon, octagon and circle. For each shape, elicit its name and as many physical characteristics as possible. List responses where appropriate.

2. Outline the five figures as floor shapes, and instruct the class that they will be directing the turtle around them.

   - Floor shapes may be made by using tape to outline them directly on the floor, drawing them on sturdy paper, or drawing them on large sheets of newsprint or newspaper.

   - Find three student volunteers: a turtle to follow the directions, a trainer to give the directions and an assistant trainer to write down directions until the procedure is correct.

   - Remind the class that, when the turtle walks, its feet are placed one in front of the other with the heel of the front foot touching the toes of the back foot.

3. How does a trainer go about starting the turtle along one side of the triangle? The trainer must determine:

   - In which direction the turtle should move;

   - the number of turtlesteps to take,

   - and, which type of turn should be made.

4. The three pupils work with input from other group members when necessary until a suitable procedure has been agreed upon, for example:

```
TO.WALK.TRIANGLE
  FD 4
  RT 120
  FD 4
  RT 120
  FD 4
  RT 120
END
```

5. Choose different groups of three students each to develop procedures for the square, the pentagon, and the octagon.

   . Does this shape look familiar? (Pupils may recognize the shape of a stop sign.) This is called an octagon.

6. Once procedures for the four shapes (triangle, square, pentagon, and octagon) have been written to walk them out, go back and add some information to the chart.

   . How many sides does a triangle (square, pentagon, octagon) have? How many turns? How is the number of sides related to the number of turns?

   . What changes in appearance do you notice from the triangle to the octagon? Which one looks more like a circle? Why do you suppose that is?

7. Discuss how the turtle walks a turtle shape. Elicit that it must move a certain amount of steps, turn, then repeat its actions until the shape has been traced.

   . Have a group attempt walking the circle.

   . At this time, the concept of forward a little, turn a little for accomplishing a circle is sufficient.

   . If you wish to add information regarding the circle to the chart, it would suffice to use "many" for the number of sides and number of turns.

Applications:

1. Take the procedures and eliminate the procedure name line. Read the procedure orally or place it on the board and have students predict outcomes.

2. Matching: On a worksheet, write the procedure instructions for the four regular polygons. This will be Column A. In Column B, draw the finished shapes. Have the students draw lines from the shape to the instructions for completing that shape.

Summary:

1. What are the five shapes we walked out today?

2. What makes an octagon unique?

3. What makes a square unique?

4. What makes a triangle unique?

5. What makes a pentagon unique?

6. How does a turtle walk a circle?

Assignments:

1. Write the procedure directions on index cards, leaving off the procedure name line. Put the procedure name line on the back of the respective card. For this activity, the cards are mixed.

   - Students work in pairs. Students take turns choosing a card. When the procedure name line comes up, the player must tell the instructions for carrying out the procedure. When the instruction side comes up, the player must predict the shape.

2. Write procedures for the two closed shapes which come between pentagon and octagon. (hexagon, 6 sides, and heptagon, 7 sides)

Assignments for More Advanced Students:

1. Make a turtle turnwheel of 360 degrees from a paper plate and a fastener or a turtle turn floor wheel large enough to accommodate a human pointer.

2. Extend collection of procedures to include dodecagons.

TOPIC 40 - E

Polygons and Their Origins

Performance Objectives - Students will be able to:

- . Visually identify: triangle, square, rectangle, pentagon, octagon.
- . Define the common properties of a polygon.
- . Make distinctions between classes of polygons identify a square as a specific kind of rectangle

Vocabulary - New: polygon, angle, rectangle
Review: triangle, square, pentagon, octagon, sides

Motivation:

1. Display a triangle, pentagon and octagon on the board, chart, or flannelboard. These are the shapes we walked out when we were "playing turtle." Which one has the shape of a stop sign? Who remembers the name of this shape?

2. Who remembers the names of any of the other shapes? What else do you know about them?

Aims: What do the members of the polygon family have in common?

Development:

1. Add a rectangle to the collection of polygons shown. What is this shape called? (If the students cannot name the figure, tell them the name and label the shape.)

   - . How does the rectangle compare to the other shapes?
   - . What is the name of each of these shapes? (triangle, rectangle, pentagon, octagon)
   - . Who can count the number of sides in a triangle?
   - . Who can count the number of sides in a rectangle?
   - . Who can count the number of sides in a pentagon?
   - . Who can count the number of sides in an octagon?

2. Develop the concept of angle. Draw a representation of a right angle on the board. I am putting two sides which point in different directions together. Do you see anything that looks like this (point) about our rectangle?

   - . Who can trace the part of our rectangle that looks like this? Is there any other part of our rectangle that looks like this?
   - . Part of the word "rectangle" can be used to tell what this piece of the rectangle is called. It can also be used to tell what these are. Draw an obtuse and an acute angle on the board. Does anyone know what these are called? Does my clue help? Part of the word "rectangle" is a clue. If the students are not able to supply the word, tell them the word.

. Who can explain what an angle is? Accept any correct age-grade appropriate response. Students, even young ones, can generally describe an angle in terms of a corner or indicate that some sort of turning motion is necessary. If answers are not forthcoming, use the following questions.

  - What did I do at the board to make the angle? Repeat the action if necessary.
  - If you were going to draw an angle with your pencil, how would you get from the first side of the angle to the second side? Allow a child to demonstrate if necessary. How many angles does each of these shapes have? Triangle, rectangle, pentagon, octagon.

3. Review: How many sides does a triangle have? How many angles?
How many sides does a rectangle have? How many angles?
How many sides does a pentagon have? How many angles?
How many sides does an octagon have? How many angles?

4. Develop the concept of a polygon as a closed shape. How is our picture of an angle different from the four shapes we have been discussing? Our four closed shapes belong to a family called polygon. What would we have to do to our angles to make them into polygons? Polygon is a hard word to say. Who would like to say "polygon" for us? Can our polygons have fewer than three sides? If not, why not? Who can count the number of sides in each of our polygons?

5. Show a square. What can you tell about this shape? Is this a closed shape? Is this a polygon? Is there another polygon that looks something like the square? The square is a special kind of rectangle. Can you explain why? Could the square be a rectangle?

Application:

1. Using at least two polygons, draw a picture of a scene or design. The number of polygons could be increased to make the task more appropriate for the group. Depending on the stage of small motor coordination, students may work with templates or rulers.

Summary:

1. What is an angle?
2. Who can point to an octagon?
3. What is a polygon?
4. Who can point to a pentagon?
5. What is a rectangle?
6. How many sides and angles does a triangle have?
7. What is a square?

Assignments for Prereaders or Readers:

1. Develop a theme drawing using one kind of polygon, e.g., a triangle.

2. Play a guessing game using objects which may be seen. Look around the room for an object which looks like it has a polygon shape on it. Give a clue, e.g. "I see something which seems to have a rectangle on it." The object could be the classroom door. The player who guesses correctly takes the next turn giving clues. Do not equate the two-dimensional polygon with the three-dimensional representation; i.e., a door is not a rectangle.

3. Play a guessing game using the vocabulary words. "I am thinking of a special kind of rectangle. Who can read my thought?" The player who guesses the word gets the next turn.

Assignments for Readers:

1. Matching - On a worksheet which has the five polygons and a representation of an angle drawn under Column A and the six accompanying labels written under Column B, have the students draw a line from each representation to the word which names it.

2. Make a polygon booklet. Develop a page for each desired polygon, using an illustration and a label.

TOPIC 40 - A

Polygons and Their Origins

Performance Objectives - Students will be able to:

- Define the properties of a polygon.
- Make distinctions between classes of polygons.
- Identify a square as a specific kind of rectangle.

Vocabulary - New: polygon, angle, rectangle

Review: triangle, square, pentagon, octagon, sides

Motivation:

- Display a triangle, pentagon and octagon on the board or chart.
- What are the names of these figures?
- What else do you know about them?

Aims: What do the members of the polygon family have in common? What makes each member unique?

Development:

1. Add a rectangle to the collection of polygons shown. What is this shape called? How does it compare to the other shapes? Do you see any similarity among parts of these four words? (triangle, rectangle, pentagon, octagon)

2. Develop the concept of angle. You noticed that "angle" appears in two of our words and that "gon" appears in two of our words. They can both be used to mean angle. What do you think an angle is? (Accept any age-grade appropriate responses. Draw a representation of an angle on the board.)

3. Develop the concept of a polygon as a closed shape. How is our picture of an angle different from the four shapes we have been discussing? Our four closed shapes belong to a family called polygon. What do you know about part of the word polygon? ("gon" = angle) "Poly" means several or many. What is the fewest number of sides our polygons have? How does each of the sides relate to the number of angles in each polygon?

4. Show a square. What can you tell about this shape? Is this a polygon? Is there another polygon that has the same characteristics? Could the square be considered a rectangle? Could the rectangle be considered a square?

Application:

1. Using several polygons, draw a picture of a scene or design. The number of polygons could be varied to make the task more appropriate for the group. Depending on the stage of small motor coordination, students may work with templates or rulers.

Summary:

1. What is an angle?
2. What is a polygon?
3. What is the relationship between the number of sides in a polygon to the number of angles?
4. What is a rectangle?
5. What is a square?

Assignments:

1. Develop a theme drawing using one kind of polygon, e.g., a triangle. This may be used to discuss equilateral triangles and those which are not.

2. Bring in old magazines. Find pictures which would lend themselves to overlaying polygons. With markers or crayons, outline the representations of polygons.

Assignments for More Advanced Students:

1. Add hexagon, heptagon, nonagon, decagon, undecagon and dodecagon to the collection of polygons developed in notebooks or on worksheets.

2. Find the meanings of the prefixes in Assignment 1 which give clues to the number of sides and angles in a polygon. Write three quiz questions for your classmates.

TOPIC 41

Polygons

Performance Objectives - Students will be able to:

- Explore and identify the following polygons: square, rectangle, triangle, and circle.
- Write commands to draw a square, a rectangle, a triangle, and a circle.
- Write a program to draw a square, a rectangle, a triangle, and a circle.

Vocabulary: Polygon, square, rectangle, triangle, circle, angle.

Motivation:

- What is a polygon?
- How many polygons can you name?
- What is the smallest number of sides needed to make a polygon?
- What instructions would you need to draw a square, a rectangle, a triangle, a circle?
- How would you write a program to draw a square, a rectangle, a triangle, and a circle?

Aim: How can we draw polygons using Logo?

Development:

1. List four different polygons.
2. List the instructions to draw the above listed polygons.
3. Write programs to draw the above listed polygons.
4. Can you write a program to draw any other polygon?

5. The programs are:

```
TO SQUARE
  REPEAT 4 [FD 50 RT 90]
END

TO RECTANGLE
  REPEAT 2 [FD 50 RT 90 FD 30 RT 90]
END

TO TRIANGLE
  REPEAT 3 [FD 50 RT 90]
END

TO CIRCLE
  REPEAT 36 [FD 4 RT 10]
END
```

Summary:

1. How can we write a program to draw the a square, a rectangle, a triangle and a circle, etc.?

Assignments:

1. Write the instructions to draw a square, a rectangle, etc.

2. Write a program to draw a square, a rectangle, etc.

3. Write a program to draw any other polygon.

TOPIC 42 - E

Using Variables

Performance Objectives - Students will be able to:

- Use the same procedure with different inputs to get the same figure in different sizes.

Vocabulary: procedure, input, variable

Motivation: How can we make the same figure in different sizes?

Aim: To use variables to draw the same figure in different sizes.

Development:

1. Here are two procedures for making triangles:

```
TO TRI50
REPEAT 3[FD 50 RT 120]
END

TO TRI10
REPEAT 3[FD 10 RT 120]
END
```

- Both are equilateral triangles, one smaller and one larger. The only difference in their procedures is the FD input.

- It would save time to make one procedure that could draw any size equilateral triangle. This can be done if we make the length of the sides of the triangle a variable rather than a constant number. This procedure shows how we do that:

```
TO TRIANGLE :LENGTH
  FD :LENGTH
  RT 120
  FD :LENGTH
  RT 120
  FD :LENGTH
  RT 120
END
```

```
TO TRIANGLE :LENGTH
  REPEAT 3[FD :LENGTH RT 120]
END
```

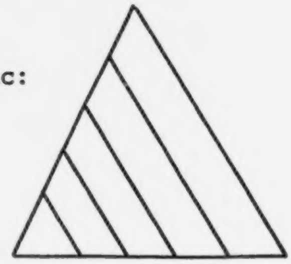
2. When your computer says: TRIANGLE DEFINED, try this:

```
TRIANGLE 50
```

. Then CLEARSCREEN and try: TRIANGLE 20, TRIANGLE 85, etc.

Application:

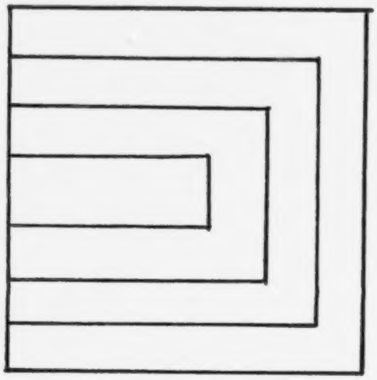
1. Using your new procedure, make this graphic:



A

2. Use this procedure for making boxes to create this graphic:

```
TO BOX :SIDE
  REPEAT 4[FD :SIDE RT 90]
END
```



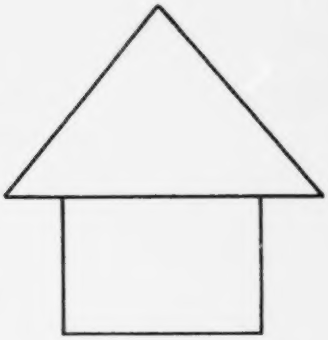
B

Summary:

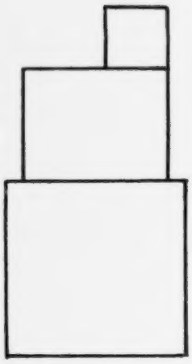
1. What is used to change the value of a line?

__Assignment__:

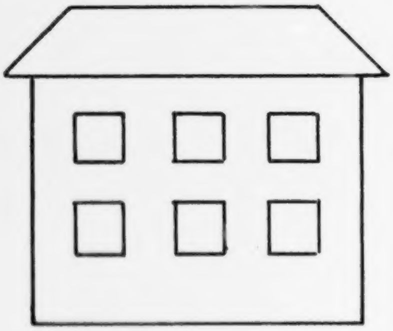
1. Use your variable procedures to make these houses:



A



B



C

TOPIC 43 - M

Variables and Polygons

Performance Objectives - Students will be able to:

- Write a program to generate a square, a rectangle, a triangle, a pentagon, and a circle.
- Write a program to generate variable sized polygons.
- Write a program to generate any regular polygon.

Vocabulary: polygon, regular polygon, square, rectangle, pentagon, triangle, circle, independent variable, dependent variable, function, angle

Motivation:

- How many polygons can you name?

- What is the smallest number of sides needed to make a polygon?

- How would you write programs to draw a square, a rectangle, a triangle, a pentagon, a circle?

- How would you write a program to generate a variable sized square, rectangle, etc.?

- How would you write a program to generate any regular polygon?

Aim: How can we generate and investigate polygons using Logo?

Development:

1. List the different types of polygons.

2. List the programs for the above listed polygons.

3. Write a program to generate any regular polygon.

4. Write a program to generate variable sized regular polygons.

5. The programs are:

```
TO SQUARE
 REPEAT 4 [FD 50 RT 90]
END

TO RECTANGLE
 REPEAT 2 [FD 50 RT 90 FD 30 RT 90]
END
```

```
TO TRIANGLE :SIDE
 REPEAT 3 [FD :SIDE RT 90]
END

TO PENTAGON :SIDE
 REPEAT 5 [FD :SIDE RT 72]
END

TO CIRCLE :SIDE :ANGLE
 REPEAT 36 [FD :SIDE RT :ANGLE]
END

TO POLY :N :S
 REPEAT :N [FD :S RT 360/:N]
END
```

Summary:

1. Can we write a program to generate all regular polygons? All polygons?

Assignments:

1. Write programs to generate a square, a rectangle, etc.
2. Write a program to generate a square, a rectangle, etc. with variable sides.
3. Write a program to draw variable sides regular polygons.
4. Write a polygon program to generate a circle.
5. Write a polygon program to generate different sized circles.

TOPIC 44 - E

Circles

Performance Objectives - Students will be able to:

- Make a circle.
- Define procedures that will make circles.

Vocabulary - Review: FD, RT, LT, REPEAT

Motivation:

- Have the students copy: FD 8 RT 10. Have them copy it again and again. Keep repeating the same commands.

- After FD 8 RT 10 has been repeated various times: What do you think the turtle is drawing?

Aim: How can we use circles to enhance procedures?

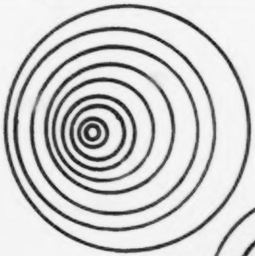
Development:

1. Copy: (Note: This topic has been prepared for Apple Logo.)

```
REPEAT 36[FD 1 RT 10]
REPEAT 36[FD 2 RT 10]
REPEAT 36[FD 3 RT 10]
REPEAT 36[FD 4 RT 10]
REPEAT 36[FD 5 RT 10]
REPEAT 36[FD 6 RT 10]
REPEAT 36[FD 7 RT 10]
REPEAT 36[FD 8 RT 10]
REPEAT 36[FD 9 RT 10]
REPEAT 36[FD 10 RT 10]
```

2. Write a procedure that will make the design made above. Call it CIRCLES.

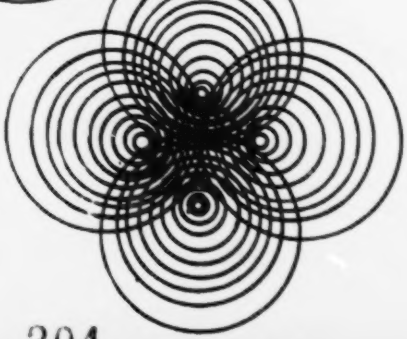
A



3. Copy: `REPEAT 4[CIRCLES RT 90]`

4. Write a procedure called MANYCIRCLES.

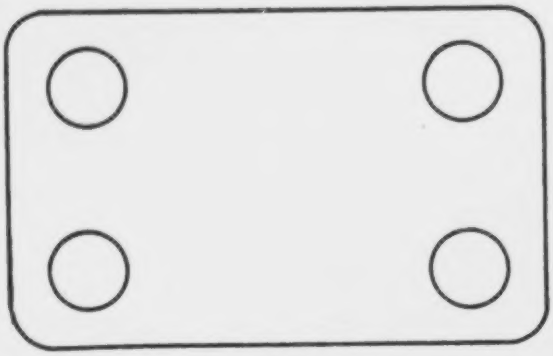
B



Applications:

1. Write a procedure that uses the circle procedure to put a circle in each of the four corners of the video monitor.

```
TO CIRCLE
  REPEAT 36[FD 4 RT 10]
END
```



A

2. Write a program that uses CIRCLE as a subprocedure.

Summary:

1. Copy the following three procedures.

```
TO X
  REPEAT 36[FD 5 RT 9]
END

TO Y
  REPEAT 36[FD 5 LT 10]
END

TO Z
  REPEAT 30[FD 5 RT 10]
END
```

2. Which procedure makes a complete circle?

3. Which procedure makes an incomplete circle?

Assignment:

1. Write a program that makes a face using circles of different sizes as subprocedures.

TOPIC 44 - A

Circles

Performance Objectives - Students will be able to:

- Make a circle.
- Define procedures that will make circles.
- Use circles as subprocedures in programs.

Vocabulary - Review: FD, RT, LT, REPEAT

Motivation:

- Copy the following: FD 8 RT 10. Copy it again and again. Keep repeating the same commands.
- After FD 8 RT 10 has been repeated various times: What do you think the turtle is drawing?

Aim: How can we use circles to enhance procedures?

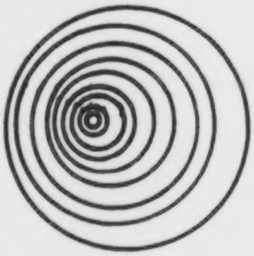
Development:

1. Copy: (Note: This topic has been prepared for Apple Logo.)

```
REPEAT 36[FD 1 RT 10]
REPEAT 36[FD 2 RT 10]
REPEAT 36[FD 3 RT 10]
REPEAT 36[FD 4 RT 10]
REPEAT 36[FD 5 RT 10]
REPEAT 36[FD 6 RT 10]
REPEAT 36[FD 7 RT 10]
REPEAT 36[FD 8 RT 10]
REPEAT 36[FD 9 RT 10]
REPEAT 36[FD 10 RT 10]
```

2. Write a procedure that will make the design above. Call it CIRCLES.

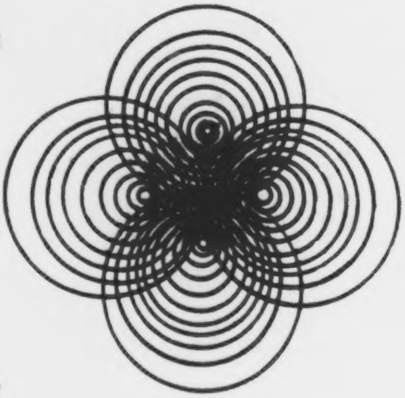
A



3. Copy: `REPEAT 4[CIRCLES RT 90]`

4. Write a procedure called MANYCIRCLES

A



5. Copy: REPEAT 360[FD 1 RT 1]

6. How can this circle be made smaller?

   . Use inputs smaller than 1. For example:

     - REPEAT 360[FD .5 RT 1]

     - REPEAT 360[FD .07 RT 1]

Applications:

1. Copy the following three procedures:

```
TO X
  REPEAT 36[FD 5 RT 9]
END

TO Y
  REPEAT 36[FD 5 LT 10]
END

TO Z
  REPEAT 30[FD 5 RT 10]
END
```

2. Which procedure makes a complete circle?

3. Which procedure makes an incomplete circle?

4. Make a "bull's eye."

5. Write a procedure that will make a "bull's eye."

Summary:

1. What are some ways of making circle procedures?

2. How are circle prcedures used as subprocedures in Logo programs?

3. When using REPEAT to make circles, what is the minimum product needed when you multiply the "number of times" by the angle?

4. How can we tell that the following command does not make a circular figure?

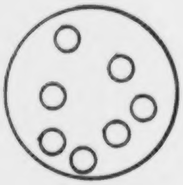
```
REPEAT 36[FD 6 RT 5]
```

<u>Assignments</u>:

1. Write a program that makes a face using circles of different sizes as subprocedures.

2. Write a program that puts small circles inside a larger circle.

A



TOPIC 45

Coordinates

Performance Objectives - Students will be able to:

- Use the command SETX to move the turtle.
- Use the command SETY to move the turtle.
- Use the command SETPOS to place the turtle in a specific location.

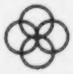
Vocabulary - New: SETX, SETY, SETPOS.
Review: PU, PD, FULLSCREEN.

Motivation:

- Copy CIRCLE and RINGS. (Note: This topic has been prepared for Apple Logo.)

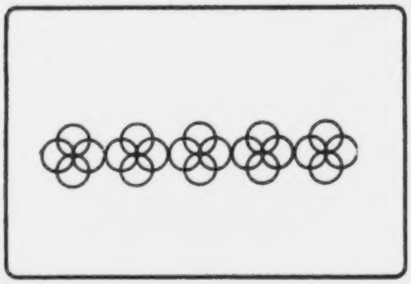
```
TO CIRCLE
  REPEAT 36[FD 2 RT 10]
END

TO RINGS
  REPEAT 4[CIRCLE RT 90]
END
```

A



- Allow pupils to experiment with RINGS. On the board draw a screen and five RINGS in a row.

B



- Using RINGS, what is one way of making this design? Allow sufficient time for investigation and experiments.

- After allowing pupils to demonstrate their solutions, introduce the concepts found in the Cartesan X and Y coordinate system.

- Demonstrate one way of making a series of five RINGS using the new commands:

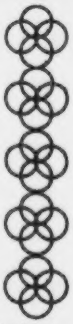
```
CS
RINGS PU SETX 45 PD RINGS
PU SETX 90 PD RINGS
PU SETX -45 PD RINGS
PU SETX -90 PD RINGS
```

<u>Aim</u>: How can we move the turtle to a specific location on the screen without using FD or BK?

<u>Development</u>:

1. Using RINGS, make a series of five vertical rings. Use the same inputs used with SETX to make the five horizontal rings.

A

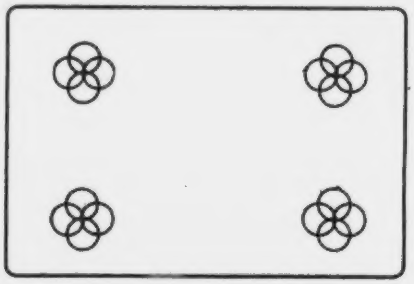


2. Introduce SETPOS. Using SETPOS, how can the previous exercise be solved?

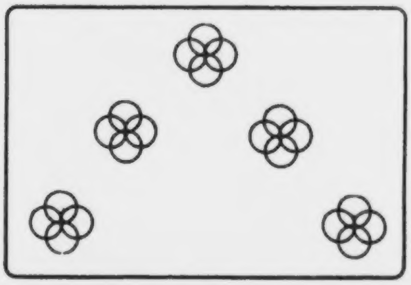
<u>Applications</u>:

1. Using the concepts introduced, write a procedure that makes each of the following:

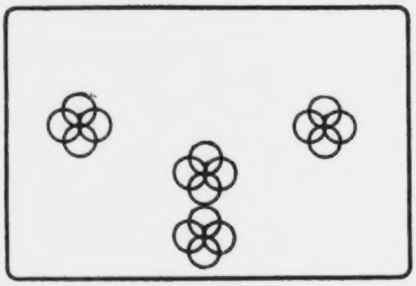
B



C



A



2. Write a program that uses SETX, SETY, and SETPOS in any of its subprocedures.

Summary:

1. How may we use SETX and SETY to shorten our programs?

2. How may we use SETX and SETY to place the turtle in different areas of the screen?

3. How can SETPOS be used to shorten programs?

Assignments:

1. Write a program that will place small drawings in the following places on the screen:

   . SETX 50 SETY 38
   . SETX 98 SETY -67
   . SETX -100 SETY 100
   . SETPOS [43 -80]
   . SETPOS [0 0]

TOPIC 46 - E

Logo and Words

Performance Objectives - Students will be able to:

- Write Logo procedures using the PRINT (PR) command and correct punctuation.
- Display and manipulate Logo words and lists on the monitor screen.

Vocabulary - New: FIRST, BUTFIRST (BF), BUTLAST, BUTLAST (BL).
Review: PRINT (PR), brackets, quotation marks (quotes), primitive

Motivation:

- Computers help us understand things more clearly by drawings, pictures and graphics. They also help us by performing difficult mathematical problems quickly.

- How can computers help us communicate with other people? (By printing out messages on the monitor or printer for others to read.)

Aim: How can we write Logo procedures to print out words and messages on the monitor?

Development:

1. Print out a single word on the screen. (Note: This lesson is written for Apple Terrapin Logo) Note placement of space and quotes after PRINT (PR). Type:

```
PRINT "HELLO
PR "MONDAY
PR "JULY
```

   - Print out your own first name on the monitor screen.

2. Print out a group of words by putting them in a Logo list. Note placement of brackets. Note also that a Logo list does not have to be a complete sentence.

```
PR [HELLO EVERYBODY!]
PR [TODAY IS MONDAY.]
PR [I LOVE NEW YORK.]
PR [RED YELLOW GREEN BLUE]
```

   - Print out your own full (first, middle, and last) name on the monitor.

   - Use your own name in a sentence. Print out the sentence on the monitor.

212

3. Parts of words can be isolated and printed separately by use of the primitives FIRST and LAST. Note placement of quotation marks. For example:

```
PR FIRST "FLORIDA
PR LAST "FLORIDA
```

4. Parts of Logo lists can also be isolated and printed separately by use of the Logo primitives FIRST and LAST. Note placement of brackets.

```
PR FIRST [FLORIDA IS FAR AWAY.]
PR LAST [FLORIDA IS FAR AWAY.]
```

5. BUTFIRST (BF) is a Logo primitive that can be used to isolate and print out all except the first letter of a word. BUTFIRST (BF) can also be used to print out all except the first item in a Logo list.

```
PR BUTFIRST "MARY
PR BF "FLORIDA
PR BF [RED BLUE GREEN]
PR BF [TODAY IS MONDAY.]
```

6. BUTLAST (BL) is a Logo primitive which can be used to isolate and print out all but the last letter of a word. BUTLAST (BL) can also be used to print out all except the last item in a Logo list.

```
PR BUTLAST "Mary
PR BL "FLORIDA
PR BL [I LOVE NEW YORK.]
PR BL [RED YELLOW BLUE GREEN]
```

7. What will be printed out when you type: `PR []` ? Type these and compare:

```
REPEAT 4[PR "HELLO]
REPEAT 5[PR [] PR "HELLO]
```

<u>Applications</u>:

1. Combine the above primitives and use them in Logo procedures to print out messages. What messages will these procedures print out?

```
TO DATES
  PR []
  PR FIRST [TODAY IS MONDAY.]
  PR BF [TODAY IS TUESDAY.]
  PR[]
  PR FIRST [YESTERDAY WAS TUESDAY.]
  PR BF [YESTERDAY WAS MONDAY.]
END
```

```
  TO GOSSIP
  PR LAST [I SEE MARY.]
  PR BF [JOHN LIKES PIZZA.]
END
```

Summary:

1. How are the following primitives used in Logo?

```
FIRST               LAST           BUTFIRST (BF)
BUTLAST (BL)        PRINT (PR)
```

- In the explanation, include correct use of punctuation (brackets and quotes.)

Assignments:

1. Write your own version of the DATES procedure that prints out the following:

```
THIS MONTH IS JULY.
LAST MONTH WAS JUNE.
```

2. Write your own version of the GOSSIP procedure. Use the names of television characters.

Assignments for Advanced Students:

1. Study the effect of the various combinations of primitives in the following procedure.

```
TO SPELLDOWN
  PR []
  PR FIRST "FLORIDA
  PR FIRST BF "FLORIDA
  PR FIRST BF BF "FLORIDA
  PR FIRST BF BF BF "FLORIDA
  PR LAST BL BL "FLORIDA
  PR LAST BL "FLORIDA
  PR LAST "FLORIDA
END
```

2. Write a similar procedure entitled COLORDOWN which will print each color in the following Logo list on a separate line:

```
[RED BROWN GREEN WHITE BLUE ORANGE]
```

TOPIC 46 - A

Logo and Words

Performance Objectives - Students will be able to:

- Write Logo procedures using the Logo primitives WORD, SENTENCE (SE), FPUT, and LPUT.

Vocabulary - New: item.
Review: SENTENCE, parentheses.

Motivation:

- In our last lesson whenever we combined two or more items into a SENTENCE, the computer always printed out the items of the sentence by placing them on two or more different lines.
- How can we write a procedure which will combine two or more items into a SENTENCE which can then be printed out all on one line?

Aim: How can we use Logo primitives to write procedures to print messages on the monitor?

Development:

1. We can instruct Logo to combine two items to make a Logo WORD. (Note: This lesson was written for Apple Terrapin Logo.)

```
PR WORD "SUN "SHINE
PR WORD "CAT "S
```

. Note the placement of the quotation marks.

2. Parentheses must be used when more than two items are combined into a single Logo WORD. If they are omitted, Logo responds with an error message.

```
PR (WORD "PEN "MAN "SHIP )
PR (WORD "BOOK "KEEP "ER "S )
```

. Note: There should be a space before the closing parenthesis.

3. We can use the Logo primitive SENTENCE (SE) to combine a Logo word and a Logo list.

```
PR SENTENCE "KITTENS [ARE CUTE.]
PR SE [MY NAME IS] "MARY.
```

. Note placement and use of both the brackets and the quotation marks.

4. Parentheses must be used to print out a Logo SENTENCE (SE) that combines three or more Logo words and lists. If parentheses are omitted, Logo responds with an error message.

```
PR (SE "I "LOVE [NEW YORK.] )
PR (SE [THE WEATHER] "TODAY [IS CLOUDY.] )
```

   . There should be a space before the closing parenthesis.

5. A new item can be added to the beginning of a Logo list by using the Logo primitive FPUT ('first-place' or 'put at the front'). Type:

```
PR FPUT "APPLE [PEAR PLUM PEACH]
PR FPUT "JULY [WAS A HOT MONTH.]
```

   . Note that quotation marks are used in front of the item to be added. Then this is followed by the list enclosed in brackets.

6. A new item can be added to the end of a Logo list by using the Logo primitive LPUT ('last place' or 'put last'). Type:

```
PR LPUT "APPLE [PEAR PLUM PEACH]
PR LPUT "JULY. [HOT DAYS COME IN]
```

   . The placement of the quotation marks and the brackets is the same as in FPUT.

<u>Applications</u>:

1. Use these primitives in a procedure which prints out a weather report. What will the following procedure print out? Note placement of all primitives and punctuation.

```
     TO WEATHER
       PR []
       PR SE FIRST [TODAY IS CLOUDY.] BF [TODAY IS NOT CLOUDY.]
       PR []
       PR  ( SE FIRST [YESTERDAY TODAY TOMORROW] FIRST [[IT WAS] [IT IS]
        [IT WILL BE]] LAST [SNOWING. FOGGY. RAINING.] )
       PR []
PR SE LAST [YESTERDAY TODAY TOMORROW]  LPUT WORD "SUN "SHINE. [WE WILL
          HAVE]
     END
```

<u>Summary</u>:

1. How are the following primitives used in Logo?

   WORD   SENTENCE (SE)   LPUT   FPUT

Assignments:

1. Write a procedure that prints out a list of five fruits. Edit the list to include the names of two more fruits at the beginning of the list.

2. Write a procedure that prints out the full names of five classmates. Edit the procedure to add two more names to the end of the list.

Assignments for Advanced Students:

1. Edit the WEATHER procedure by combining the commands differently to create a new weather report.

2. Edit the WEATHER procedure to report smog for yesterday, bright sun for today and wind for tomorrow.

3. Edit the WEATHER procedure to include the appropriate names of the days of the week (Monday, Tuesday, etc.)

TOPIC 47

Writing with Logo

Performance Objectives - Students will be able to:

- Write procedures which print out short messages in standard memo style.

Vocabulary - New: memo.
Review: hyphen, edit, revise.

Motivation:

- If we wanted to use the computer to leave messages for others to read, how could we get Logo to help us?

Aim: How can we use Logo to write a memo or short note?

Development:

1. Use a friend's name as the title of a procedure in which a short message is printed out. (Note: This lesson was written for Apple Terrapin Logo.) The name of the person sending the message should be typed in first.

```
TO MARY
  PR [FROM SUE]
  PR [DO YOU LIKE COMPUTERS?]
  PR [I WILL CALL YOU LATER.]
END
```

2. Edit the message to include a line with the date and time that the message was written. A line could also be added which repeats the title so that the reader will know that the message is really for them.

```
TO MARY
  PR [TO MARY]
  PR [FROM SUE]
  PR [JUNE 4, 10:30 AM]
  PR [I WILL CALL YOU LATER.]
END
```

3. Write a longer message to your friend. Make your message easier to read and understand by:

   - Skipping a line where it seems appropriate.

   - Not splitting words at the ends of lines.

   - Finishing a phrase before going to a new line.

. Type:

```
TO MARY
  PR [TO MARY]
  PR [FROM SUE]
  PR [JUNE 4, 10:30 AM] PR[]
  PR [DO YOU LIKE COMPUTERS]
  PR [AS MUCH AS I DO?]
  PR[]
  PR [I WILL CALL YOU LATER.]
  PR [FROM MY HOUSE AS SOON AS]
  PR [I GET HOME FROM SCHOOL.]
```

Applications:

1. Write procedures that print out a memo for each of three or four friends in your class. Your friends can call up their messages by typing their names on the computer keyboard.

2. The messages can also be saved on a diskette, or printed out in hard copy on a printer.

Summary:

1. How do we use Logo to write messages on the computer?

Assignments:

1. Write a procedure that prints a message to your teacher in which you list the titles of all the procedures you defined today.

Assignments for Advanced Students:

1. Write a procedure that prints out the instructions on how to use Logo to draw a square.

TOPIC 48

Naming

Performance Objectives - Students will be able to:

- Write Logo procedures in which the name of a variable is changed with the MAKE command.

Vocabulary - New: MAKE.
Review: Variable.

Motivation:

- Almost everything we can think of can be described or named in several ways. For example, the names Bill, Billy, Willy, Will, William, Billy-boy, etc. could all be used to call one particular individual who would answer to all of those names.

- How can we do this with the words and numbers that we work with in Logo?

Aim: How do we write a Logo procedure in which the name of a word or a number is changed?

Development:

1. Use the MAKE command to shorten the name of a lengthy word. (This lesson was written for Apple Terrapin Logo) Type:

```
TO SPEED.TYPE
  MAKE "TV "TELEVISION
  PR :TV
END
```

2. Use the MAKE command to name a number or an arithmetical result. Type:

```
TO CHANGE.NUM
  MAKE "FWD 50
  ST
  REPEAT 4[FD :FWD RT 90]
END

TO CHANGE.NUMS
  MAKE "FWD 25 + 25
  MAKE "RGT 60 + 60
  ST
  REPEAT 3[ FD :FWD RT :RGT
END
```

Applications:

1. One way to use MAKE is in joining words or parts of words. Type:

```
    TO REWRITE
      MAKE "T "RALPH
      MAKE "S 2
      MAKE "R "YES
      MAKE "Q "WALDO
      MAKE "P 4
      MAKE "O "TER
      MAKE "N "EMERSON
      MAKE "M 6
      MAKE "L "DAY
      PR (SE :T :Q :N)
      PR :S :P :M
      MAKE "K (WORD :R :O :L)
      PR :K
    END
```

Summary:

1. How does the MAKE command change the names of Logo words and numbers?

Assignments:

1. Write a procedure using the MAKE command in which initials are used to print out the names of five friends.

Assignments for Advanced Students:

1. Write a procedure using the MAKE command in which the titles and categories of different types of books in a library can be printed out.

TOPIC 49

Logo Conversations

Performance Objectives - Students will be able to:

- Edit a procedure which simulates a conversation with the computer.

Vocabulary - New: converse, choice, response, top level, branching, keyboard character
Review: READCHARACTER (RC) STOP

Motivation:

- Students are responsive to use of branching procedures which offer different options in a conversational style as the program is run.

- How can we combine procedures in order to simulate a chat with the computer?

Aim: If you could talk to the computer, how could you make it draw a simple picture for you?

Development:

1. A simple conversation can be started by offering a two-part choice, for example:

   Do you want to draw a boat?
   Answer yes or no.

2. We can write a Logo procedure for this. (Note: This lesson was written for Apple Terrapin Logo.)

```
TO CHOOSE
  PR [DO YOU WANT TO DRAW A BOAT?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
END
```

3. Edit this procedure to instruct Logo to read whatever keyboard character READCHARACTER (RC) is pressed.

   - Logo is then instructed to make a new name for that keyboard character. In this procedure the new name is KEY. Edit:

```
TO CHOOSE
  PR [DO YOU WANT TO DRAW A BOAT?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  MAKE "KEY RC
END
```

4. Edit this procedure to direct Logo to perform some action depending on which key is pressed. Note correct placement of punctuation.

```
TO CHOOSE
  PR [DO YOU WANT TO DRAW A BOAT?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  MAKE "KEY RC
  IF :KEY = Y PR [OKAY. WHERE'S MY PEN?]
  IF :KEY = N PR "GOODBYE!
END
```

5. Branching programs should include a Logo instruction [i.e. STOP] which will return Logo to top level. This is useful in two ways:

   . When a keyboard character is pressed which the procedure does not designate, an error message is avoided.

   . When all branching is complete, the program ends. Otherwise, it needs to be stopped with CONTROL G.

```
TO CHOOSE
  PR [DO YOU WANT TO DRAW A BOAT?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  MAKE "KEY RC
  IF :KEY = Y PR [OKAY. WHERE'S MY PEN?]
  IF :KEY = N PR "GOODBYE!
  STOP
END
```

   * N.B. Some implementations of Logo require brackets around STOP - i.e. [STOP].

<u>Applications</u>:

1. Combine a subprocedure which draws a boat with the conversational procedure above. The picture will be drawn if the 'Y' key is pressed.

```
TO BOAT
  HOME CS HT PD
  LT 45 FD 50
  RT 135 FD 100
  RT 135 FD 50
  RT 45 HOME
  PU FD 35 PD
  FD 80 RT 120
  REPEAT 3[FD 50 RT 120]
END
```

. Edit:

```
TO CHOOSE
  PR [DO YOU WANT TO DRAW A BOAT?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  MAKE "KEY RC
  IF :KEY = Y BOAT
  IF :KEY = N PR "GOODBYE!
  STOP
END
```

Summary:

1. How did we write a procedure which seemed to permit us to talk with the computer?

Assignments:

1. Write a procedure which draws a simple house. Edit the CHOOSE procedure so that the computer offers to draw a house instead of a boat.

Assignments for Advanced Students:

1. Edit the CHOOSE procedure so that if the 'N' key is pressed, Logo asks: "Would you prefer to draw a house?" Then write a graphics procedure which draws a house. Edit CHOOSE again and insert your new 'house' procedure as another option.

TOPIC 50 - E

Introduction to Sprites

Performance Objectives - Students will be able to:

- Explain what sprites are.
- List the qualities of sprites.
- Create sprites on the screen.

Vocabulary: sprites, CARRY

Motivation:

- What are sprites?

Aim: To use our computers to see sprites.

Development:

1. The turtle has some friends called sprites. But we can't see them unless we make up what they look like.

2. What are some shapes that could be used for video games?

   - Elicit the following shapes that sprites can take: plane, rocket, truck, ball, box.

3. We have to tell a sprite what shape it is. Each sprite has a number. We'll talk to Sprite 1 to tell it what to be. How should we do this?

4. Let's first tell it where it should be on the screen. Let's use the HOME position that the turtle used.

   ```
   TELL SPRITE 1
   HOME
   ```

   - What happened? Why didn't anything appear on the screen?

   - What other information do we need to give the sprite? (Elicit the concepts of shape and color).

5. What shape should the sprite be? (Have students select one of the five shapes offered in step 2).

   ```
   TELL SPRITE 1
   HOME
   CARRY :PLANE
   ```

   - What effect does the word CARRY have on the sprite? (Sprites have no shape until you have them CARRY one).

   - What else is needed to see the sprite? (Color)

6. Let's give the sprite a color.

```
TELL SPRITE 1
HOME
CARRY :PLANE
SETCOLOR :RED
```

7. When we were playing with our friend the turtle, what did we have to do to clear the screen of his drawings?

   . One way to remove a sprite is to set its color to 0.

   . Color chart

   CLEAR
   BLACK
   GREEN
   LIME
   BLUE
   SKY
   RED
   CYAN
   RUST
   ORANGE
   YELLOW
   LEMON
   OLIVE
   PURPLE
   GRAY
   WHITE

<u>Applications</u>:

1. Recreate Sprite 1 with another color.

2. Type in FD 20. What happens?

<u>Summary</u>:

1. What must be done in order to see a sprite?

<u>Assignments</u>:

1. Experiment using the FD, BK, RT, and LT turtle commands to move the sprite across the screen.

2. Experiment with changing sprite's color and shape.

TOPIC 50 - A

Introduction to Sprites

Performance Objectives - Students will be able to:

- Explain what sprites are.
- List the qualities of sprites.
- Create sprites on the screen.

Vocabulary: Sprites, heading, direction.

Motivation:

- Can something exist if it cannot be seen?

Aim: To create objects that can be seen and set into motion.

Development:

1. Let's talk about our friend the turtle. What is he? What can he do?

2. The turtle has some friends called sprites. But we can't see them unless we make up what they look like.

3. What are some shapes that could be used for video games?

   - Elicit the following shapes that sprites can be: plane, rocket, truck, ball, box.

4. We have to tell a sprite what shape it is. Each sprite has a number. We'll talk to Sprite 1 to tell it what to be. How should we do this?

```
TELL SPRITE 1
```

5. Let's first tell it where it should be on the screen. Let's use the home position that the turtle used.

```
TELL SPRITE 1
HOME
```

   - What happened? Why didn't anything appear on screen? What other information do we need to give the sprite?

6. What shape should it be? (Use whichever of the five shapes students select.)

```
TELL SPRITE 1
HOME
CARRY :PLANE
```

- What sense does the word CARRY make? (The sprite should CARRY the shape indicated after the dots.) What else is needed to see the sprite?

7. Let's give it a color. We can use either SC or SETCOLOR.

```
TELL SPRITE 1
HOME
CARRY :PLANE
SETCOLOR :RED
```

- The sprite should now appear in the center of the screen. (Note: A chart of the sprite colors should be displayed.)

CLEAR
BLACK
GREEN
LIME
BLUE
SKY
RED
CYAN
RUST
ORANGE
YELLOW
LEMON
OLIVE
PURPLE
GRAY
WHITE

- Tell Sprite 2 to be a blue rocket.

- Tell Sprite 3 to be a purple truck.

- Create your own sprites.

<u>Summary</u>:

1. How did we get the sprites to show movement? And what do we call this movement?

<u>Assignments</u>:

1. Play with the rocket to take off at different speeds and land.

2. Create a red ball that moves around.

3. Experiment with different shapes, colors and speeds.

TOPIC 51 - E

Making Sprites Move

Performance Objectives - Students will be able to:

- Explain how to give a sprite speed.
- Explain wrapping.
- Create sprites that move and wrap around the screen.

Motivation:

- Review what sprites are, what they look like and what they can do.

Aim: To make sprites move over the screen.

Development:

1. Tell Sprite 1 to be a green rocket.

```
TELL SPRITE 1
HOME
CARRY :ROCKET
SETCOLOR :GREEN
```

2. What can we do to get our rocket to move? (Set the speed of the rocket).

```
SETSPEED 127 (This is the fastest the sprites can go.)
OR SS 127
```

3. How do we make a slower rocket?

```
SETSPEED 1 OR SS 1
```

4. How do we get the rocket to continue to move without typing in many instructions?

5. Tell your rocket to move slowly.

6. Gradually increase the speed of the rocket.

7. We've noticed that our rocket disappears and comes back at the opposite end of the screen. This is called wrapping (like gift wrap goes around.)

8. What would happen if we gave the rocket the speed of zero? (It would stop.) Why?

Summary:

1. What are sprites?

2. How can we get sprites to move at different speeds?

Assignments:

1. Play with the rocket to take off at different speeds and land.
2. Create a red ball that moves around.
3. Experiment with different shapes, colors and speeds.

TOPIC 52-A

Making Sprites Move

Performance Objectives - Students will be able to:

- Explain how to give a sprite speed.
- Explain how to reverse a sprite's direction.
- Explain wrapping.
- Create sprites that move and wrap around the screen.

Motivation:

- Does moving sprites make them animated?

Aim: How do sprites move?

Development:

1. Tell Sprite 1 to be a green rocket.

```
TELL SPRITE 1
HOME
CARRY :ROCKET
SETCOLOR :GREEN
```

2. What could we do to get our rocket to move? (Set the speed of the rocket.)

`SETSPEED 127 OR SS 127` (this is the fastest the sprite can go.)

3. How do we make a slower rocket?

`SETSPEED 1 OR SS 1` (This is the slowest the sprite can go.)

4. Tell your rocket to move slowly, faster, faster, faster.

5. How can we get the rocket to go backwards? (use negative speeds.)

`SETSPEED -127 OR SS -127` (This is the fastest it can go backwards.)

`SETSPEED -1 OR SS -1` (This is the slowest it can go backwards.)

6. We've noticed that our rocket disappears and comes back at the opposite end of the screen. This is called wrapping (like gift wrap goes around.)

7. Why does the rocket wrap around the screen?

8. What would happen if we gave the rocket the speed of zero? (It would stop.)

9. Tell your rocket to go forward, stop, and then go in reverse.

```
SS 100
SS 0
SS -100
```

. Play with the rocket to take off at different speeds, land, go into reverse, etc.

Summary:

1. Now that we can make sprites move, can we say that they are animated?

Assignments:

1. Create a red ball that flies around.

2. Create an orange box that moves slowly.

3. Experiment with different shapes, colors and speeds.

TOPIC 53 - E

Giving Sprites Direction

Performance Ojectives - Students will be able to:

- Explain how the sprite moves.
- Give sprites different directions.

Motivation:

- What do you already know about sprites?

Aim: To be able to give sprites different directions.

Development:

1. How did we get the turtle to move in different directions? (We turned it a specific number of degrees.)

2. Let's play the game Sprite H.

   - Make the following chart and place it in the front of the room.

```
        360

   270        90

        180
```

   - Make a circle with the above being the four directions on the outer edge. Explain that 360 is the amount needed to make a complete circle.

   - Have the students face in the direction that they want the sprite to go.

3. What heading would we give a sprite to tell it to move

   - straight ahead? (SETHEADING 0)

   - straight right? (SETHEADING 90)

   - straight down? (SETHEADING 180)

   - straight left? (SETHEADING 270)

4. Tell Sprite 1 to be a blue plane that goes to the right slowly.

```
TELL SPRITE 1
CARRY :PLANE
SC :BLUE
SS 50
SH 90
```

Applications:

1. Change Sprite 1 by deciding first on its shape, color, speed, heading. Write it down. Then type it on the computer to see if the sprite does what you thought it would.
2. Experiment by having your sprites go at different headings and speeds.

Summary:

1. By setting down coordinates we have been able to give sprites direction. Can we now say that sprites are animated?

Assignment:

1. Have students write riddles for sprites.

TOPIC 53 - A

Giving Sprites Direction

Performance Objectives - Students will be able to:

- Explain how the sprite moves.
- Give sprites different directions.

Vocabulary:

Motivation:

- Do we now have animation?

Aim: To help students understand what directions are to sprites.

Development:

1. How did we get the turtle to turn so it could move forward in a different direction? (We turned it a specific number of degrees.)

2. Sprites move in a similar way, but we call their direction heading. How would we tell them in what direction to head? (SETHEADING)

3. Judging from what we know about the turtle's movement, what heading would we give a sprite to tell it to move straight right?

```
SETHEADING 90 OR SETHEADING :EAST (SH 90 OR SH :EAST)
```

4. How would we make it go straight up?

```
SETHEADING 0 OR SETHEADING :NORTH (SH 0 OR SH :NORTH)
```

5. How would we make it go straight down?

```
SETHEADING 180 OR SETHEADING :SOUTH (SH 180 OR SH :SOUTH)
```

6. How would we make it go straight left?

```
SETHEADING 270 OR SETHEADING :WEST (SH 270 OR SH :WEST)
```

7. What happens with all the numbers from 0 to 360 that we haven't used? (These vary the angle of the sprite's direction or heading.)

8. Let's tell Sprite 1 to be a blue plane that goes east slowly.

```
TELL SPRITE 1
CARRY :PLANE
SC :BLUE
SS 50
SH 90
```

9. Create Sprite 2 by deciding first on its shape, color, speed, and heading. Write it down. Then type it on the computer to see if the sprite does what you thought it would.

   . Create more sprites.

Summary:

1. Have we been able to make sprites animated?

Assignments:

1. Students should create task cards for each other. The object is to create the sprite described on the task card.

TOPIC 54-E

Many Sprites at Once

Performance Objectives - Students will be able to:

- Create several sprites.

Motivation:

- What do you think will happen if we move more than one sprite at the same time?

Aim: How do we get more than one sprite on the screen at once?

Development:

1. The number of sprites available depends on the computer you are using. Each sprite needs its own number and set of commands. For example,

```
TELL SPRITE 1        TELL SPRITE 2
CARRY :PLANE         CARRY :ROCKET
SC :RED              SC :BLUE
HOME                 HOME
SS 50                SS 100
SH 0                 SH 90
```

2. How can we get the sprites to stop moving? FREEZE

3. FREEZE the sprites and change their colors.

   - Did they both change color? (Since you were last speaking to Sprite 2, only the rocket changed color.)

Summary:

1. Now that we have several sprites on the screen at one time, can we do more with them?

Assignments:

1. Play with creating as many sprites as are available on your computer. Freeze them. Change their color, speed, shape, and heading.

TOPIC 54-A

Sprites and More Sprites

Performance Objectives - Students will be able to:

- Create several sprites.
- Move several sprites at the same time.

Motivation:

- Can we use our new knowledge of sprites to make animated graphics?

Aim: How do we use our knowledge of sprites to create graphics?

Development:

1. The number of sprites available depends on the computer you are using. Let's start by creating two sprites who will play.

2. What should they be? Example: One red plane that travels north slowly and one blue rocket that travels east quickly.

```
TELL SPRITE 1          TELL SPRITE 2
CARRY :PLANE           CARRY :ROCKET
SC :RED                SC :BLUE
HOME                   HOME
SS 50                  SS 100
SH 0                   SH 90
```

3. Where are the sprites? (They both begin at the home position, so Sprite 1 covers Sprite 2, but both are there and can move the way we tell them to.

4. How can we get a sprite to stop moving? (FREEZE)

5. How can we get them to move again? (THAW)

Summary:

1. Having made many sprites move at the same time, can we do more with sprites?

Assignments:

1. Play with creating as many sprites as are available on your computer. Freeze them. Thaw them. Change their headings.

2. Play with the sprites using turtle commands. Use FD, RT, LT, BK with a number input.

3. Play dodgeball with a truck and a ball.

TOPIC 55 - E

Making New Shapes

Performance Objectives - Students will be able to:

- Locate and control the Makeshape keys.
- Learn the command Makeshape.
- Learn to use the shift key.
- Create shapes in Makeshape mode.
- Learn the command Makecharacter.

Vocabulary: makeshape, cursor

Motivation:

- How do you think the computer people made the ball, rocket, and other shapes?

- Use Makeshape to get up each shape. Explain: You can make shapes, too.

Aim: What advantages are there in being able to command Makeshape?

Development:

1. Practice typing in the command Makeshape.

2. What is the function of the cursor?

3. Have students number the columns of their grids from 1 to 16.

4. Practice recognizing the keys and their functions by flashing the cards.

5. Send students to the computer to operate the keys so that the cursor blackens its column. Then operate the keys so that the column is left unshaded.

    - What causes the shading of the column?

    - What causes the unshading of the column?

    - Practice shading and unshading. Do shading and unshading of the rows.

6. Can you get a completely round shape? Type in MS :BALL. Is it completely round? Why not?

7. Type in Makecharacter numbers to show the letters of the alphabet.

Applications:

1. Practice Makecharacter to see the use of makeshape. Students will draw their own shapes in grids and practice putting them in Makeshape.

Summary:

1. What command is needed to make new shapes?

2. In what ways is the writing of programs made better with the use of the command Makeshape?

Assignments:

1. Have the students create a bigger ball.

SECONDARY GRADE TOPICS

TOPIC 1

Computers In Society

Performance Objectives - Students will be able to:

- List familiar places in our society that use computers.
- Identify items evidencing use of computers in these places.
- Describe our society as one that makes good and abundant use of computers.

Materials:

- As many of the following items as possible:

  computer printed grocery receipt
  computer printed phone/utility bill
  bank teller window receipt
  ads for computer appliances
  computer magazines
  credit card
  airline ticket
  concert ticket
  digital watch
  punch cards

Motivation:

- Imagine a spaceship from another planet landing on earth; two creatures come out and begin exploring. All they find is this box of items. (Take out box of items.) They begin examining the materials.

Aim: How are computers used in our society?

Development:

1. Have students identify the items. Discuss where they have seen them and who might use them.

2. Ask the students how the aliens can use these items to learn how people live.

3. What can the aliens learn about us and our lives based on these items? (the jobs we hold, entertainment we like, our personalities, our intelligence, etc.) What conclusions might the aliens make about a computer?

4. Form a list of places in our society where the use of a computer is important to the function of that place:

   Stores
   Industries/Factories
   Telephone Company
   NASA/Communication Satellites
   Computer-dating services
   Banks
   Other Businesses
   Hospitals
   Libraries
   Airports

Summary:

1. Many areas of our lives are affected by computers.

2. Computers have helped improve many of the services we use.

Assignments:

1. Class trips to any of the above places.

2. Collage or scrapbook of computerized items.

3. Choose someone from the past. How do you think this person would react to computers if he/she were alive today? How do you think this person might have benefitted from using a computer. Example: Martin Luther King might have used a computer to help him write his speeches.

4. Can you solve this riddle? "How did the bride and groom computers who just got married first meet each other?" (Through COMPUTER DATING!!)

TOPIC 2

Future Computers

Performance Objectives - Students will be able to:

- Identify the important characteristics of modern computers.
- Make predictions of future computers.

Vocabulary: translator, electronic, artificial intelligence, videodisk, telecommunication, robot, speech synthesizer, fifth generation.

Motivation:

- Ask the students what they would like computers to do.
- Discuss robots and what role they will play in the future.
- Discuss video games and the sound effects they have.
- Discuss Epcot Center in Florida.
- Have students share their reports on fifth generation computers

Aim: What will computers be like in the future?

Development:

1. Demonstrate various capabilities of present day computers.
   - A game program illustrates sound and graphics.
   - A clock program demonstrates the ease with which a computer can exercise loops.
   - An arithmetic program illustrates accuracy and speed.
   - A decision-making program using the IF/THEN statement based on data the computer has been given demonstrates logic capability.
   - A database program illustrates information storage and retrieval.
2. Have the class read current articles from newspapers and magazines which describe recent advances in computer technology.
3. Describe the development of videodisks, each of which can store more information than five thousand floppy disks.

4. Introduce the concept of artificial intelligence in computer programs.

   - Computers that play chess.
   - Computers that ask patients questions and diagnose their illnesses.
   - Computers that find error patterns in students' answers to math problems.
   - Computers that recognize shapes in television pictures.
   - Computerized appliances that make housework easier.
   - Robots that perform boring, repetitive jobs.
   - Computers that translate from one language to another.

5. Introduce the concept of a cashless, paperless society brought about by:

   - Telecommunication from a home terminal connected to terminals in businesses;

   - electronic mail which does not need paper or postage,

   - electronic bank tellers that take deposits, cash checks, and pay bills,

   - information networks which serve as paperless encyclopedias.

Applications:

1. List tasks in your home which could be done by computerized appliances or robots. Substantiate the choices.

2. List all the ways in which paper could be saved if correspondence were done electronically instead of by mail.

3. In the future how might household furnishings differ from what you have in your home today? Based on new computer technology, what new things would you buy? What things might you discard? Give reasons.

Summary:

1. How will computers be different in the future?

2. How will advances in computers affect our society?

Assignments:

1. Write a play, dialogue or composition describing an encounter with a fifth generation computer.

2. Draw pictures or construct models to illustrate what future computers may look like.

3. Write a description of a family which can work, shop and play without ever leaving the house because of a computer terminal. Would such a life be desirable?

4. Debate the question: "Computers will make life more interesting in the future."

TOPIC 3

The Computer as a Problem-Solving Tool

Performance Objectives - Students will be able to:

- Understand that a computer is a tool that can help people think.
- Recognize the limitations of a computer in problem solving.
- Appreciate the unique capabilities of a computer.

Vocabulary: program, database, simulation, tool.

Motivation:

- Tie a number of books together so that the load is heavy. Ask a pupil to lift it. Now place the books on one end of a lever (a large board with a chair as a fulcrum). Ask the individual to lift the books with the aid of the lever. Which was the easier way to lift the books? Why?

- Establish the concept that there are tools to help people with problems.

Aim: How does a computer help us in solving problems?

Development:

1. The lever is a tool to help people lift heavy objects. Name some other tools that make physical work easier.

2. The computer is a tool that has been designed to help people use their ability to think.

    - Without a human being to use it, the computer is like a drill or a comb without someone using it. Each accomplishes nothing without a human being, but each helps a human accomplish things more easily and quickly.

    - People must tell computers exactly what to do and how to do it.

3. Computers are 100% obedient but very "stupid." They know nothing except what a human has told them.

    - They must be told exactly what to do, in detail.

    - They must be told the exact order in which to do each thing.

    - Except for special programs which use Soundex algorythms, they cannot figure out what you write by sounding it out. Everything you tell the computer to do must be spelled exactly right.

4. Instructions to the computer are written as programs in a computer language. They must be written correctly for the computer to operate.

   - Since a computer always follows the instructions, it doesn't make mistakes. "Computer Error" is usually "Programmer Error."

5. A computer was designed to solve problems.

   - It can do things very fast. Its speed is one of its most important assets. It can do thousands of addition examples in a blink of an eye.

   - It can store and recall thousands of facts instantaneously.

   - It can repeat a series of steps over and over without errors.

   - It can compare and sort information based on the data and instructions given it.

   - It can show many possible outcomes based on different combinations of data.

6. Mathematical computation, database management, and simulation design are computer capabilities which assist humans in problem solving.

Applications:

1. Describe a situation in which the computer's mathematical capabilities would be preferred to a human's.

2. Describe a situation in which the speed of a computer's recall of stored memory would be of great help in problem solving.

3. Write a list of instructions for "a man from Mars" to tell him how to brush his teeth. Include every detail and be certain to place each step in its correct order.

Summary:

1. What qualities of a computer make it an outstanding problem solving tool?

2. What role must a human being play in making a computer useful?

3. What tasks could we not expect a computer to do?

Assignments:

1. People blame computers for many mistakes. Plan an advertising campaign using posters, slogans and jingles which will tell the public about the good things computers can do.

2. Very complicated problems can be broken down into simple parts. Think of a problem you have. Explain it to a computer, step by step, in a logical order.

TOPIC 4

The Effect of Computer Design on Problem-Solving

Performance Objectives - Student will be able to:

- Understand the form in which a computer's information is stored.
- Understand that a computer is designed with math logic inside.
- Understand how information is displayed.

Vocabulary: binary system, ASCII Code, bit, byte, ROM, circuit.

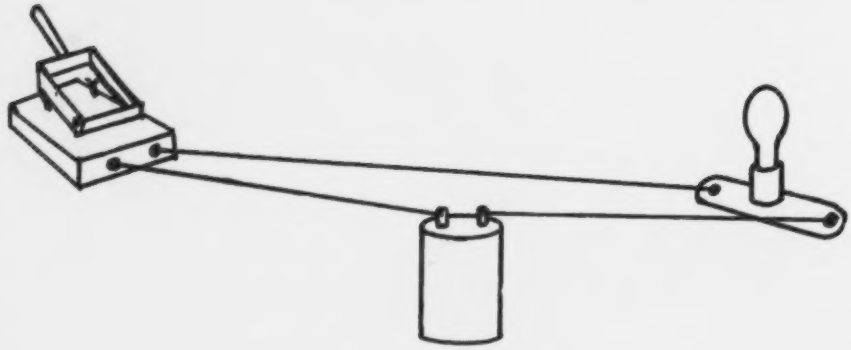
Motivation:

- Show the students two combs: one with most of the teeth missing; the other in good condition. Which is the better comb for doing the needed job? Why? (Relate function and design.)

Aim: How does the design of a computer make it an excellent tool for problem solving?

Development:

1. The design of the two combs in our experiment illustrate how the design of a tool can affect its performance. The same is true with the computer.
   - Its speed and memory capacity make it very useful in problem solving.
   - The ability to break down complex operations into very simple steps simplifies the actual programming process.
2. All the computer's information must be stored as numbers. These numbers stand for a pattern of electrical current.
3. Electricity can only be in two forms, off or on. Demonstrate with a simple circuit utilizing a switch, three wires, a bell or a bulb in a socket and a dry cell, how electricity is on and off.



(A)

- Different combinations of the position of the switch will cause the bulb to light in different patterns.

- Demonstrate appliance: on, off,on, off, on, on, off, on
  on, on, on, off, off, on, off, on etc.

4. Each key on the keyboard is assigned a number based on an electrical pattern.

   - There are only two numbers used: a zero which stands for electricity off and a one which stands for electricity on.

     - A number system that uses only zeros and ones is called a binary system (bi meaning two as in bicycle or bisect).

   - Each zero and each one is called a bit. Eight bits make a byte. A byte equals one character or keystroke.

   - When a key is pressed, the computer changes the keystroke into the coded number for that key. For example, the lowercase letter "a" may have the coded number 65. The computer stores a string of zeros and ones which are the binary equivalent of 65.

   - When the computer prints to the screen or printer, it changes the string of zeros and ones back into the character.

   - Since electricity travels very fast and the circuits on chips are very small these changes are instantaneous, hence the speed of computer performance.

5. Make a copy of the following chart and post in the room to demonstrate the different binary combinations.

THE ASCII CODE

A	0100 0001	J	0100 1001	S	0101 0011
B	0100 0010	K	0100 1011	T	0101 0100
C	0100 0011	L	0100 1100	U	0101 0101
D	0100 0100	M	0100 1101	V	0101 0110
E	0100 0101	N	0100 1110	W	0101 0111
F	0100 0110	O	0100 1111	X	0101 1000
G	0100 0111	P	0101 0000	Y	0101 1001
H	0100 1000	Q	0101 0001	Z	0101 1010
I	0100 1001	R	0101 0010		

6. All of the computer's operations are carried out by special circuits.

   . Since most computers were originally used for calculations, a program for math logic is in the ROM (Read Only Memory) of the computer.

   . Because of this, the programmer does not have to instruct the computer how to add, subtract, multiply or divide. These instructions were given "back in the factory."

Applications:

1. What is your computer's code for letters and characters? Consult the manual which came with the computer.

2. Make up a riddle and write the answer using the ASCII Binary Code. See if your friends can translate the answer into English.

3. Do the following examples in each of three ways. Have a friend time how long it takes you to get the 4 answers by hand. Then time how long it takes to get the 4 answers by the hand calculator. Finally, time how long it takes the computer.

BY HAND	BY HAND CALCULATOR	BY COMPUTER
93 - 27 =	93 - 27 =	PRINT 97 - 27
84 + 26 =	84 + 26 =	PRINT 84 + 26
27 x 3 =	27 x 3 =	PRINT 27 * 3
84 / 6 =	84 / 6 =	PRINT 84 / 6

Summary:

1. In what form is computer's information stored?

2. How does a computer display information on the monitor or printer?

3. How does the design of the computer contribute to its speed of operation?

4. How does the computer know how to do math?

Assignments:

1. Create a different number system for a computer to use.
2. What could power a computer other than electricity? What would be the advantages and disadvantages?
3. How did the changes in computer design since 1944 provide for faster performance?

TOPIC 5

Computer Disadvantages - Job Security

Performance Objectives - Students will be able to:

- Define technology.
- Explain how improved technology sometimes causes unemployment.
- Explain why computer use is increasing daily.
- List the kinds of jobs computers are likely to replace.

Vocabulary: technology, unemployment, generations, computer technology, electronic mail, ENIAC, microchip, automation.

Motivation:

- Bring in the Sunday Times employment ads and initiate a discussion of their purpose.
- Point out that in all the hundred of ads there aren't any for blacksmiths, ice men, chimney sweeps or stove stokers. Explain that such jobs have been replaced by machinery.
  - Discuss what kind of machinery has replaced each of the examples.

Development:

1. What is meant by technology?
   - Technology is the application of scientific principles to solve practical problems, in other words, the use of machinery to do the work of men. The automobile, the wheel and the pulley are examples of technology.
   - Technology has both positive and negative effects on society. It makes life easier, but it also often replaces many jobs. Refrigerators did away with the need for ice men, and dishwashing machines replaced dishwashers.
2. Why is computer technology increasing at such a rapid pace?
   - Since the first electronic computer (ENIAC) was introduced in 1946 by John Mauchly and Presper Eckert, the history of computers has been one of increasing speed and diminishing size and costs. With the development of the integrated circuit chip, the microchip, in the 1970's, computers have become much smaller and much less expensive. The result of this has been a dramatic increase in the number of computers in use in our society.
   - In 1980 there were about 700,000 computers in use. By the end of 1983, there were approximately 13,000,000 in use.

3. What kinds of tasks are computers good at?

   . Given enough resources and time, people can do anything computers can do. In fact, people outperform computers in most sophisticated decision-making processes. The only advantage that computers have over people is their incredible speed. Mathematical operations, sorting, selecting, transmitting and receiving data, printing and certain kinds of decisions are the tasks computers can do very quickly.

4. The increased use of computer technology will have an effect on what kinds of jobs?

   . Jobs which can be done more efficiently by a computer will be the most directly affected.

     - The position of file clerk may disappear since computer files are much more efficient than paper and cabinet files. A database management software package could eliminate the need for an entire department.

     - A computer system equipped with spreadsheet software could eliminate the jobs of many bookkeepers.

     - Automatic teller machines have already substantially reduced the need for human tellers.

     - The coming of electronic mail--the transfer of messages via computer terminals--may threaten the survival of the mail clerk's job.

   . Computer technology has even entered industry. More and more industries are using robotic arms to relieve workers of many monotonous and sometimes dangerous tasks.

     - They are now routinely used to put goods in cartons, paint cars and remove products from furnaces.

   . With the use of computer technology, work is more productive. Unfortunately, this productivity means that people lose jobs.

     - Office workers no longer have to spend hours retyping letters because word processors remember and reproduce documents at will. Therefore, the demand for typists has been reduced.

     - In the publishing industry, computerized equipment has eliminated the jobs of typesetters and pressmen. Many people believe the laser printer will even eliminate the need for print shops since people will be able to use the laser printer to produce their own stationery and newsletters.

     - As computers get more and more "intelligent", their applications will alter employment in many other areas.

- If computerization does not eliminate a job, it is likely to change it in some way. Even in the professions of law and medicine, computer skills will be needed to access a data base or use diagnostic equipment. Teachers will routinely use computers and educational software to teach and reinforce skills.

- Within the next ten years computer will be involved in 80% of the jobs in our society.

5. What can be done to prevent unemployment caused by computer technology?

   - One answer to unemployment caused by computerization is training and retraining. Workers must be willing to learn about new and different technologies; it will not go away by ignoring it. Students preparing to enter the work force should acquire computer skills as well as those of the particular field for which they are training.

     - For example, anyone now training to become an accountant should be sure to acquire some knowledge of computers to enable him to deal with the computerized accounting systems being used.

Applications:

1. Interview parents and other family members to discover how computers have eliminated, changed or had no effect on their jobs. Report back to class or make chart showing the findings.

2. Write future want ads for traditional jobs and speculate on how the job may be changed by computerization.

3. Organize a "Computer Career Day." Invite parents to come in to discuss their jobs and how computers affect it.

4. Use the help wanted ads and collect ads for jobs that are computer-related. Make a collage or other display of the collection.

Summary:

1. In what ways has computer technology caused unemployment?

2. What can be done to help people get jobs in the computer industry?

TOPIC 6

Disadvantages of Computers - Depersonalization and Dependency

---

Performance Objectives - Students will be able to:

- Define the terms depersonalization and dependency.
- Describe the way people become computer dependent.
- Describe the depersonalization that sometimes occurs with computerization.

Vocabulary: depersonalization, dependency, computerization, technological.

Motivation:

- Dependent vs. independent--What does each of these words mean?

- Are you dependent or independent? In what ways are we a little of each?

Development:

1. In what ways is society becoming "computer-dependent?"

   - Aside from the large number of appliances, calculators and other devices that are powered by microchips, computers are helping to regulate and control many areas of our lives. They are involved in the production of our automobiles, as well as the printing of our daily newspaper. Our stores, banks, hospitals and the U.S. Government would be hard-pressed to keep records without computers. Traffic lights, telephone systems, weather satellites and even our national defense systems are monitored and regulated by computers.

   - Although we personally may say, "I don't depend upon a computer," a quick inventory of the things, people and organizations that we depend upon will show that they depend upon computers. So in a very real sense we are computer-dependent as well.

2. Is there anything wrong in being a "computer-dependent" society?

   - One problem inherent with depending upon anyone or anything is that you may be disappointed or inconvenienced because the expected service was unfulfilled. For example, if you depend upon a bus to go to school and the bus breaks down, you'll probably be late.

   - Like anything else, computers fail. Because they provide vast numbers of people with some extremely vital services, a computer breakdown can result in very large inconveniences or serious damage.

     - To be prepared is probably the best defense against computer failure. People, banks, stores, schools or whatever should depend upon computer technology, but they must have alternate provisions if the technology fails. We should depend upon computers but we should not forget how to be independent.

3. In what ways has increased computer use added a depersonalization to our society? Depersonalization refers to the process in which a person is treated more as a thing than a person.

   - Depersonalization began during the Industrial Revolution when workers were treated solely as a means to a producing a product and a profit. The rise of unionism and reform went a long way to stopping some of the harsher sides of this process.

   - With the computerization of large parts of our society, some people believe a new wave of depersonalization has set in. There is the feeling that companies, schools, hospitals, or any kind of organization that has computerized its records and administration are relating to people less as people and more as data items in a computerized file.

   - Prior to computerization, records and the relationships that they dealt with were much more people-oriented. Organizations kept files and dealt with people on a personal basis.

   - With computerization much of this has changed. Your local druggist, for example, doesn't need to remember you and your allergy to certain medications; your record is easily accessible. The net effect is that you may feel people know about you but they don't know you. You are not thought of as a person, only as a element in a data file -- depersonalization.

Summary:

1. In what ways have we become too dependent upon the computer?

2. What do we mean by "depersonalization?" How does this affect our daily lives?

Assignments:

1. Make lists of businesses and organizations in the community that have become too computer dependent, that is, couldn't operate without them.

2. After a discussion, write a short story entitled, "The Day All the Computers Quit." Imagine what would occur in a computer-free world. Would it be better or worse?

3. Create a chart with three columns as follows:

   NO LOSS SOME LOSS GREAT LOSS

   - Research and list under each column businesses that have had some, no, or a great loss in personal service as a result of computerization.

4. Conduct a poll. Work with students to develop a questionaire aimed at getting opinions concerning computer dependency and the impact of computers on the personal touch in various fields. Distribute to neighbors and parents. Tabulate and analyze results.

TOPIC 7

Software Storage and Management

Performance Objectives - Students will be able to:

- Identify modes of storing information.
- Describe the proper way to handle tapes and insert in cassette player.
- Describe the proper way to handle diskettes and insert in disk drive.

Vocabulary - New: diskette, floppy disk, cassette tape, read only memory (ROM), random access memory (RAM), interface, hard disk, storage device.
Review: memory, central processing unit, firmware, hardware, software, disk drive.

Motivation:

- When you do your homework assignments, what instruments, objects or materials do you use so that it can be handed in, saved, and corrected?

- If we want to save or store information needed for the computer we have certain objects called storage devices available, depending upon the type of microcomputer and/or hardware we have.

Aim: How do we store and manage software?

Development:

1. Define ROM and RAM

   - Microcomputers have two types of memory: read only memory (ROM) and random access memory (RAM).

     - ROM is the permanent memory built into the microcomputer at the factory and contains all the firmware necessary to process information entered into the computer. It tells the computer how to add, subtract, print out information, etc., depending upon what we type into the computer.

     - RAM, on the other hand, is the storage location for new information given by the computer user by typing it in on the keyboard. This memory is not permanent. Any information typed into RAM is lost when the microcomputer is turned off.

     - To avoid losing the information, there must be a way to save or store it. There are many advantages of saving and storing information for future use. Try to think of some (the saving of time and effort by not having to re-enter information via the keyboard).

2. The three types of storage devices available for microcomputer systems include cassette tapes, floppy disks, and hard disks.

. A cassette tape is the most inexpensive storage device. To utilize cassette tapes with your microcomputer, you need a cassette player (tape recorder) and a special connection called an interface to be able to communicate with the computer.

- Cassette tapes have certain disadvantages. They are slow and somewhat unreliable. It is also usually hard to pinpoint needed information using fast forward or reverse. You might be familiar with this problem if you ever tried to locate a song on a music cassette.

. Floppy disks, also called diskettes, are more accurate and faster. Information is recorded on the surface of a round, flat disk. Disks are 3-1/2, 5-1/4 or 8 inches in diameter, and are permanently sealed in a cover to protect them from dust, fingerprints, and dirt. To use a floppy disk, you must have a disk drive. When the disk is inserted and activated in the disk drive, it rotates very rapidly. Locating information on a disk is much like locating a song on a record where the needle can be placed on the precise spot, without wasting time.

. Hard disks are used with sophisticated microcomputers (such as the IBM PC XT) and are not usually available for classroom use. The hard disks hold much more information than the floppy disks and cost much more.

3. Handling cassette tapes and disks

. Cassette tapes should be handled carefully, and touching the tape which might be somewhat exposed should be avoided.

- Cassette tapes are inserted in a cassette player by opening the player door (by usually pressing an "eject" key) and inserting the cassette so that it locks into the two "rollers" that advance the tape. Be sure to close the door before starting the machine.

. Disks should be handled very carefully by the label. Never touch any part visible through the openings in the protective cover because this might damage the disk.

- Before inserting the disk, make sure the IN USE light on the front of the drive is off. Never remove a disk while this light is on because you can damage the disk or destroy information.

- To insert a disk, lift the small door on the front of the drive. If there is a disk inside, remove it carefully and put it in the protective envelope. If there is no disk in the drive, hold the disk with your thumb on the label, remove it from its envelope and insert it in the drive gently.

- It should be inserted so that the oval-shaped cutout end goes in first, label side up. Do not force or bend the disk. Take it out if you notice any resistance. When the disk is completely inside, close the door.

Applications:

1. Practice loading software in the form of cassette tapes and/or diskettes (depending upon availability).

2. Determine whether the following sentences are true or false:

   . A cassette tape can hold more information than a diskette. (F)

   . Diskettes are used to store information temporarily held in RAM. (T)

   . Diskettes are much slower and less reliable than cassette tapes. (F)

   . Different microcomputer hardware components communicate with each other along interface connections. (T)

   . ROM and RAM are part of a computer system's cental processing unit. (T)

   . Firmware is stored in ROM. (T)

Summary:

1. What is the difference between ROM and RAM?

2. What are the different ways of storing or saving information?

Assignment:

1. Locate the following vocabulary words in the cryptogram. You may find the words beginning from top-to-bottom or bottom-to-top, vertically and diagonally, or, right-to-left or left-to-right, horizontally.

   diskette, floppy disk, cassette, read only memory, RAM, interface, CPU, firmware, hardware, software, storage, disk drive, hard disk

```
Z T H S R H A R D W A R E
S H A T V Q U A I K P E R
N T R O Q U N M S R N A A
E T D R U T V I K K I D W
R E D A E T D T E N V O T
A S I G F Y Q V T P S N F
W S S E P D I E T N Y L O
M A K P R R R B E A L Y S
R C O C D F D E H F Q M R
I L S K A G I P C P U E F
F K S C J R Q U V U M M T
L I E X S F B T N W T O V
D Y H D E G Z J L C A R O
I R S Q U K T P N O M Y T
```

2. Ask a student to show the class how to hold and insert a cassette tape (depending upon availability).

3. Ask a student to show the class how to hold and insert a diskette.

TOPIC 8

Introduction to the Keyboard

Performance Objectives - Students will be able to:

- Recognize the placement of letters, numbers and symbols on the keyboard.
- Recognize and know the use of special keys.
- Be able to tell the difference between a typewriter and computer keyboard.

Vocabulary: Key, keyboard, keyboarding, shift, return, control key, reset key

Motivation:

- Project a large blank keyboard using an overhead projector or display a large blank keyboard chart.

- Ask the class "Where have you seen something similar to this outline?"

Aim: What do we find on a computer keyboard?

Development:

1. Distribute a duplicated copy of the keyboard of the computer the students will be using. Project an overhead transparency or have a very large copy of this keyboard fastened to the board.

   - Have the students point to each letter of the alphabet. Tell the students that each letter is printed on a key and that what they are looking at is called a keyboard.

   - Point out that this is one of many different keyboards that are used on typewriters and computers.

     - How are words made from this keyboard? How are sentences made using this keyboard?

   - Pressing the keys to enter words or numbers into a computer is called keyboarding. This word replaces the word typewriting because the same process is now done on machines other than typewriters.

2. Have the students locate various keys on the keyboard.

   - Explain that the space bar is used to put a space between words so that sentences can be made.

   - Have students find the special punctuation marks that could end a sentence (, . ? ! : ;).

- Have the students find the numerals 1 - 0. Point out to them that there are special characters also printed on the keys.

- Ask the students to identify each symbol and tell its use. Make a list on the board showing the symbol and its use.

- Depending upon the computer you are using tell the students about the special keys on the computer and their special uses.

  - Shift key changes from capital to small letters

  - Return key takes what has been typed and puts it into the computer's memory.

  - Arrow keys moves the cursor around the screen.

  - Other keys should be taught as the need arises.

3. Some activities that could be used for drill and reinforcement include the following.

   - Draw or collect pictures to paste on the large copy of the keyboard. This would then be used as a bulletin board.

   - Keyboard Bingo. Distribute blank bingo cards. Have the students fill in any 24 letters or symbols (the middle box is pre-printed "free".) Point to a key on the large picture of the blank keyboard. If a student has that symbol, put a large X in the box. Whoever gets five keys in a row (including the free box) wins.

   - Word Spell. Prepare in advance words on index cards and place them in a box. The student picks a word out of the box and has to point to the letters on the keyboard.

   - Computer Sounds. Point to the blank keyboard and call on a student. The student must make the sound of an object represented by the blank space that was pointed to.

   - Have the students color in the letters that correspond to their name. Different colors could be used for each name.

Applications:

1. Count out a hundred words in a story book and see which letters are used most often.

2. List the symbols on your keyboard, and tell which jobs would find the symbol most useful.

Summary:

1. What is a keyboard?
2. How many letters are there on the keyboard?
3. How many numbers are there on the keyboard?
4. What is meant by keyboarding?
5. List the special keys on your keyboard and tell the use of each.

Assignments:

1. Write a report about other types of keyboards such as the Dvorak keyboard.
2. Build a model of a computer showing the keyboard you were taught.

WORKSHEET #1 - A BLANK BINGO-TYPE WORK SHEET - see rest of this file.

Student's Name______________________ Date____________

KNOW YOUR KEYBOARD

Below you will see 25 boxes. Place a letter, number or symbol in each box. Do not put anything in the box marked FREE.

When your teacher points to one of the keys on the chart of the keyboard, see if you have this symbol on your sheet. If you did, then write a big X in the box with that symbol.

When you have five in a row raise your hand. The row can go across the page; up and down; or from corner to corner.

		FREE		

TOPIC 9

The Micro as a Word Processor

Performance Objectives - Students will be able to:

- Understand the difference between a typewriter and a word processor.
- Identify the usefulness of word processing in the world around us.

Vocabulary - New: word processor, highlighting, search, replace

Motivation:

- Have the students write a short composition or letter. As they are doing this, the teacher should go around the room correcting any mistakes.

- After an appropriate amount of time ask the class, "What would you have to do to hand in a perfect paper?"

Aim: How is a word processor different from a typewriter?

Development:

1. Point out that any mistakes would have to be corrected and the composition or letter rewritten until it were perfect. Ask the class if there is an easier way.

   - If a typewriter and the use of a special correction paper or liquid is suggested, point out that the solution is still messy and time-consuming.

2. Introduce the word processor as a computer program that allows them to change what they wrote and print it out in a neat form.

   - There are many different word processors for every type of computer.

   - Note: You should be familiar with your computer keyboard and a word processing package for your machine. Carefully read the manual that came with the program so that you can relate this lesson and future lessons to the program you selected.

3. Describe several tasks a word processor can do that a typewriter can't.

   - A word processor shows the work on a monitor before it is sent to a printer.

   - All of the words that are spelled wrong can be changed on the screen.

   - If you leave out a word or sentence, the word processor will let you put it in without ruining the rest of the work.

   - You can save your work, and then bring it back at a later date to change it or add to it.

- A word processor allows you to move the words and sentences around easily.

- It can print several copies of the same report without retyping.

- Charts and tables are easily prepared with a word processor.

- Highlighting or printing words darker can be done with most word processing programs.

- You can search for and replace any word. For example, if you spelled the word "receive" wrong, you can easily search for it and correct it.

5. Develop a chart showing some of the differences between a typewriter and a word processor.

JOB	TYPEWRITER	WORD PROCESSOR
Put letters on a page	YES	YES-with printer
Correct wrong letters	YES-with special material	YES
Put in words that you left out	NO	YES
Change the order of words	NO	YES
Save your work	NO	YES
Bring back the work you saved	NO	YES
Use different type styles	YES-with special parts	YES
Move words around the page	NO	YES
Print several copies	YES-with carbon paper	YES
Charts/Tables easily made	NO	YES
Highlighting of words	NO	YES
Search and replace	NO	YES

6. Who would find a word processor useful?

- Elicit that businesses send out many of the same letters with just the names changed.

- Lawyers need to change only small sentences in standard legal forms.

- Newspapers and magazines use word processors to lay out the page neatly and to send it over phone lines to the printers.

- Teachers use word processors to make up tests and reading lessons.

- Students use these programs to hand in reports.

- Parents can use word processors to send letters to all of their friends and relatives.

7. Why would a business spend thousands of dollars for a microcomputer and a word processor?

   . Word processing saves much time. A secretary gets the job done faster on a word processor than on a typewriter.

   . When a business saves time, it also saves money.

   . Many business documents can be kept on small computer disks and not take up a lot of space.

Applications:

1. Survey the class to see if the students have computers and word processors.

2. Bring in one of the school's microcomputers and briefly demonstrate a word processor.

3. Make a bulletin board display showing the differences between a typewriter and a word processor.

Summary:

1. Why is a word processor better than a typewriter?

2. How has word processing changed the business world?

3. How is a word processor useful in everyday life?

Assignments:

1. Research and write a report on the history of word processing.

2. Where are word processors used in your neighborhood? Students should contact local businesses to see if (and how) they are using word processing.

TOPIC 10

Knowledge of a Word Processor

Performance Objectives - Students will be able to:

. Boot a word processing program.
. Understand the different subsystems of a word processor.
. Gain a working knowledge of a word processor.

Vocabulary: boot, format, editor, input, text, files, filer, store, saved

Motivation:

. Demonstrate a word processing package.

Aim: How do we make our word processor work?

Development:

1. Inserting the computer disk into the disk drive and turning the machine on is called booting the program.

   . Depending on the number of microcomputers available, have each student boot the program.

   . The students could work in small groups, with each child spending some time on the machine.

   . Note: Since there are many types of microcomputers, and many word processing programs for each kind of computer, it is imperative that you become familiar with your machine and program and adjust the following lessons accordingly.

3. What do you have to do when you want to write a composition?

   . Elicit the following:

   - You need a pencil and paper.
   - You decide on the margins.
   - You begin writing.

   . Explain that a word processor works in the same way.

   - You boot the computer program.
   - You set the margins.
   - You begin writing.

4. After you boot the program, you should enter the FORMAT part of the program.

   . The format of a page tells the computer how the paper looks after you set the margins.

   . Ask a student to measure a standard sheet of paper. (8 1/2" by 11")

   . Word processors usually print six lines per inch. How many lines can we get on a page that measures 11 inches long? (66 lines)

     - If you use legal paper (14 inches), the page is 84 lines long. (14 x 6)

   . A word processor has many type styles but it usually prints 10 letters per inch across the page. Elicit that an 8 1/2" wide paper could print 85 letters across the page.

     - For a one-inch left margin, how many spaces must be left?

     - For one-inch margins on both sides, how many letters could be typed?

     - The line spacing can also be set to double space in the format section.

5. The actual writing is done in the EDITOR section of the word processor program.

   . Have the students write (input) the heading on the paper.

   . Next have the student write several sentences.

     - The teacher should go around the room pointing out any mistakes. Do not, at this point, explain how to correct them. If the students ask, tell them they will have a chance to correct their papers in the next lesson.

   - Explain that what each student has written is called text.

6. What would happen to the text if the computer were shut off?

   . Demonstrate that the text is lost.

   . Then elicit that text must be saved before turning off the machines. Equate saving text on a computer with saving compositions in a notebook.

7. Explain that text, like paper, is stored in files in the FILER section of the word processing program.

   . Files must have a name so that they can be stored or saved on a disk.

   . Have the students save the work that they created.

Applications:

1. When working in groups, have the students talk to each other on the computer. This is a running dialogue similar to a play.
2. Write a poem.
3. Give the students the first line of a story and let them complete it.
4. Have the students write a new and modern ending to an old fairy tale.
5. Have the students write a story about a class trip or science experiment.
6. Write a letter requesting something special.

Summary:

1. How do we start our word processing program?
2. What do we have to do to set up our page?
3. Where do we enter text in a word processor?
4. How do we prevent the text from being lost?

Assignments:

1. Plan a class newspaper or subject magazine.
2. Write a "Famous People" booklet. The subjects for this writing assignment can be school or local neighborhood people.
3. Write letters to elected representatives.

TOPIC 11

Logo Closed Figure

Performance Objectives - Students will be able to:

- Draw a variety of closed figures.
- Become familiar with the sizes of 30, 45, and 90 degree angles.
- Use their knowledge of how to walk and turn their own bodies to determine how to move the turtle in order to create a desired figure or shape.

Vocabulary: right angle, graphic, PENERASE (PE), PENUP (PU), PENDOWN (PD)

Motivation:

- How can we create a variety of closed figures?

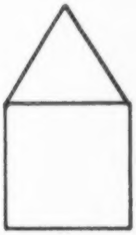
Aim: To use Logo primitives to create closed figures.

Development:

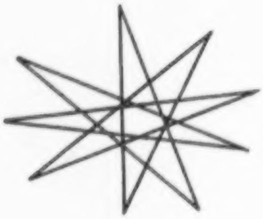
1. Plan as a group how you will move the turtle to draw the word DOG. Use the FD and BK commands with your choice of numerical inputs. Use the RT and LT commands with inputs limited to 30, 45, and 90 degrees, and their multiples.
2. Record the moves on the chalkboard as the group determines them.
3. Follow that by trying out the commands on the computer. Remember to use the PU command when you don't want the turtle to leave tracks.

Application:

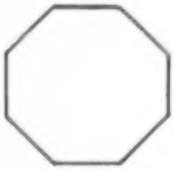
1. Try making these shapes with the turtle:



A



B



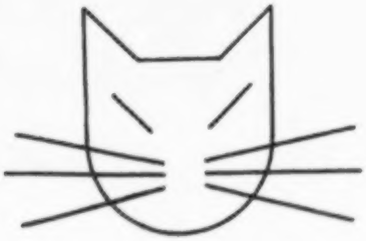
C

Summary:

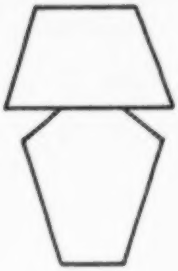
1. What effects do the primitives PE, PU, and PD have?

Assignment:

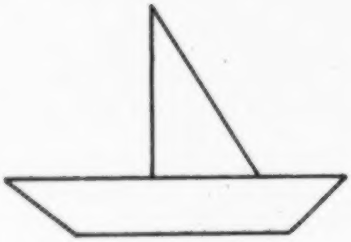
1. Try some more:



A



B



C

TOPIC 12

Logo Closed Figure and Procedures

Performance Objectives - Students will be able to:

- Use Logo commands to create a variety of closed figures.
- Develop ability to estimate correct angles needed to produce certain shapes.
- Use well developed sense of direction to select the correct commands for these graphics.
- Make procedures that will produce regular closed figures.

Vocabulary: regular figure, procedure, subprocedure

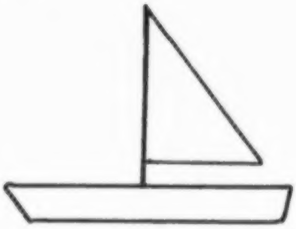
Motivation:

- How can we develop procedures for making regular closed figures?

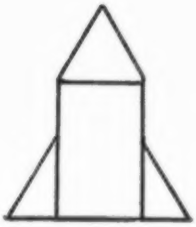
Aim: To create closed figures with Logo primitives.

Development:

1. Try making these graphics on the computer:



A



B

- As you draw your figure, keep a record on paper of the commands you used to make them.

2. When your drawing is complete, you can teach the computer to draw your sailboat and/or your rocket with one command, by making a procedure for each one. Here's how:

   - Type: TO BOAT
     - You may use any other word or letter as the title for your procedure if you wish.
   - Then press RETURN.
     - This symbol will appear on the screen:>.

- Type the commands that you used to draw your sailboat. Each time you press RETURN, the > symbol will appear on the next line. When you have typed in all of your commands, press RETURN.

- On the last line, type END. When you press RETURN, the computer will tell you: BOAT DEFINED.

- Now, if you want the computer to draw your boat, all you need to do is type the title word BOAT.

3. You can teach the computer procedures for drawing some regular figures that may be useful in making other graphics. Teach your computer these procedures:

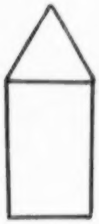
```
TO SQUARE                        TO SQUARE
  FD 40 RT 90                         REPEAT 4 [FD 40 RT 90]
  FD 40 RT 90                    END
  FD 40 RT 90
  FD 40 RT 90
END


TO TRIANGLE                      TO CIRCLE
  REPEAT 3[FD 40 RT 120]              REPEAT 36[FD 8 RT 10]
END                              END
```

Summary:

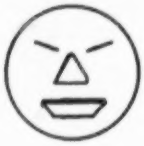
1. How can you use Logo to make regular closed figures?

Assignments:

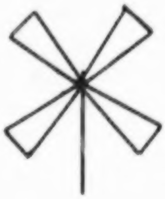
1. Make the following graphics on your computer. Write a procedure for each one. Use any of the above procedures that will help as subprocedures.



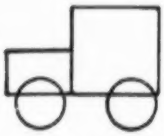
A



B



C



D

TOPIC 13

Edit Mode

Performance Objectives - Students will be able to:

- Understand and use edit commands.
- Write complex procedures in edit mode.
- Debug procedures.
- Remove procedures from the turtle's dictionary.
- Understand the difference between a subprocedure and a superprocedure.

Vocabulary: New: bug, debug, subprocedure, superprocedure
Review: procedure

Motivation:

- What are some of the errors a person may make in writing a Logo program? (syntax errors, wrong commands, etc.)
- What can be done to make finding and fixing mistakes easier? (Organizing procedure writing carefully and learning methods to fix those errors.)

Aim: How can edit commands be used to write complex programs and to debug programs?

Development:

1. Review editing commands:
    - Have students do WORKSHEET 1 (see editing command worksheet for middle experience for answers).
2. Have students experiment which of these commands will work in both immediate (or direct) mode and which will work only in edit mode. They can do this by writing commands in both modes. Have them write down their observations. (CTRL N, CTRL P, CTRL O, CTRL C and CTRL G can only be used in edit mode. The rest could be used in either.)
3. Give out and go over new editing command list. Provide time for practice using these commands.
    - Have students write and edit simple procedures using all editing commands that they have learned.
    - Have students write errors into the procedures for square, triangle or rectangle and then use editing commands to debug those procedures.

4. Writing complex procedures:

   . Elicit from the students the definition, uses and parts of a procedure.

   . Ask the following questions:

     - If you were to build a model of a boat, how could you organize the pieces to make the job easier? (You could put pieces that go together in groups and build one part at a time.)

     - How might this make it easier to fix any mistakes you made? (It would be easy to locate the problem.)

     - If you wanted to write a program for making a boat on the computer, how could you make the job easier? (Write different pieces or subprocedures for each part of the boat.)

     - Elicit from the students the advantages of writing complex procedures with separate pieces or subprocedures - easier to locate bugs (mistakes) and to edit them.

   . Ask the following question:

     - If you wanted the turtle to make the boat more than one time how could you do that without writing the subprocedures over and over? (Write all the subprocedures under one superprocedure, like boat.)

5. Have students do Worksheet 2.

6. Making debugging easier -- introduce TRACE/PAUSE

   . Explain that the TRACE command in Terrapin/Krell is particularly useful in debugging. When TRACE is ON and the procedure is executed, the turtle obeys each command and states which command is being obeyed. This helps to pinpoint the exact problem. The PAUSE command in Apple Logo is also helpful in debugging.

7. Have students do Projects 1-5.

8. Removing procedures from the turtle's dictionary:

   . Explain to students that the turtle's memory is erased each time they turn off the machine. Another way to erase the turtle's dictionary in the memory of the computer is to give the command:

   ERASE ALL (TERRAPIN/KRELL)　　　　ERALL (Apple)

     - This command erases the turtle's workspace and makes the turtle forget everything it learned. If the student wants to erase one procedure then type ERASE followed by the name of the procedure.

WORKSHEET 1

WRITE DOWN THE EDITING COMMANDS THAT WILL DO THE FOLLOWING.

1. Move cursor back one position and erase the character to the left.
2. Move the cursor forward one position without erasing a character.
3. Move the cursor back one position without erasing a character.
4. Delete the character under the cursor.
5. Delete text from the cursor position to the end of the line.
6. Move cursor to the next line.
7. Move cursor to the previous line.
8. Open a space to insert a line.
9. Exit editor and add procedure to the dictionary.
10. Exit editor without defining a procedure.
11. When you want to save a procedure, you type.
12. If you want to reload any old procedure that you saved, you must type.

WORKSHEET 2

John wanted to make a kite on the computer and he forgot to write separate procedures for each part. He needs a top, a triangle, a bow, and a tail. Please help him organize his kite by figuring which commands go under each subprocedure name, and then typing each subprocedure under a new procedure called KITE.

```
FD 60
RT 60
RT 50 FD 40 BK 40 LT 50
FD 30
RT 99
RD 30
RT 60
FD 60
RT 141
RT 19.5
FD 80
BK 80
REPEAT 3 [FD 10 RT 120]
TRI
LT 150
TRI
REPEAT 30 [FD 1 RT 1]
REPEAT 30 [FD 1 LT 1]
```

(Answers: These are in order)

TO TOP
goes from FD 60 to BK 80 (that is the first 11 lines of commands)

```
TO TRIANGLE
REPEAT 3 [FD 10 RT 120]

TO BOW
TRI
LT 150
TRI

TO TAIL
REPEAT 30 FD 1 RT 1
REPEAT 30 FD 1 RT 1
```

Projects

1. Create your own scene. Make a house with a yard. The yard may have trees, flowers, a fence, etc. Let your imagination go wild. Remember to write subprocedures for all the parts of your house scene.
2. Create a program that writes your name in print or script.
3. Create a program that draws your favorite animal or person.
4. Create your own robot.
5. Write a program that will make the U.S. flag.

Editing Command List 2

More Commands to Help Correct Typing Errors (in both immediate and edit modes)

1.	SPACE (Terrapin and Apple)	Moves all characters to the right of the cursor, including the character under the cursor.
2.	CTRL A (Terrapin and Apple)	Moves the cursor to the beginning of the line.
3.	CTRL M (Terrapin and Apple)	Same as RETURN
4.	CTRL Y (Apple)	Returns what you have killed with CTRL K
5.	ED (Terrapin and Apple)	Brings back the last procedure in EDIT mode. Note you do not have to write the name of the procedure if you want to edit the last one you had in edit mode.

COMMANDS TO CONTROL SCREEN IN IMMEDIATE (DIRECT) MODE

1.	CTRL F (Terrapin); L (Apple)	Gives full graphic screen
2.	CTRL S (Terrapin and Apple)	Gives mixed text/graphic screen (4 lines of text at bottom of screen.
3.	CTRL T (Terrapin and Apple)	Gives full text screen

TOPIC 14

PENUP and PENDOWN Commands

Performance Objectives - Students will be able to:

- Change the turtle's position on the screen without leaving a trail.
- Resume drawing with the turtle when it is in the desired position.
- Create graphics that involve a shape within a shape.
- Begin projects with the turtle in any position on the screen.

Vocabulary: PENUP (PU), PENDOWN (PD), PENERASE (PE), HOME, SETHEADING (SETH)

Motivation:

- How can we move the turtle to various places on the screen?

Aim: To draw figures without leaving a line.

Development:

PENUP (PU)	Lifts the turtle's pen up so that when it moves it does not draw lines.
PENDOWN (PD)	Puts the turtle's pen down so that it draws a line as it moves.
PENERASE (PE)	Commands turtle to erase any lines it moves over. This can be used to make corrections in drawings
HOME	Moves turtle to the center of the screen.
SETHEADING (SETH)	Turns the turtle so that it is heading in the direction indicated by the input of degrees following the command. For example, SETH 0 or SETH 360 sets the turtle heading north; SETH 90 sets turtle heading east; SETH 180 sets turtle heading south; and SETH 270 sets turtle heading west.

1. Using the basic commands: FD, BK, RT, LT, and the commands in the vocabulary list above, try to draw your initials on the screen.

   - To move the turtle without leaving a trail, be sure to press PE before typing in the commands. Likewise, press PD before typing the commands for the lines you want drawn.

2. Try this game for practicing the PU and PD commands:

TIC-TAC-Logo

. Try this program to make the frame:

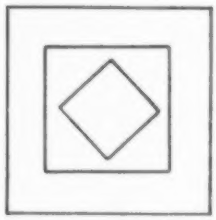
```
TO T
  PU LT 90 FD 70
  PD BK 150 RT 90
  PU FD 50 LT 90
  PD FD 150 BK 50 RT 90
  FD 50 BK 150 RT 90
  PU FD 50 LT 90
  PD FD 150
  PU HOME
```

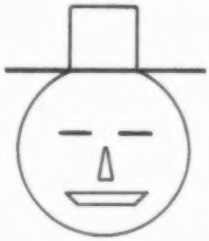
. Try these procedures for making the markers:

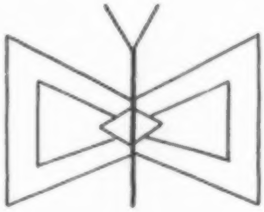
```
TO X
  LT 45 FD 20
  BK 10 RT 90
  FD 10 BK 20
END

TO O
  REPEAT 36[FD 1 RT 10]
END
```

Application:

1. Using the PU and PD commands, try drawing these figures:


A


B


C

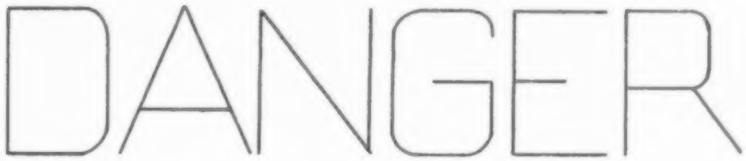
Summary:

1. How do the PENUP and PENDOWN commands facilitate the drawing of figures?

Assignments:

1. Compose a picture that requires starting with the turtle in a position other than HOME.

2. Use Logo to draw your name, or a sign. For example:



A

TOPIC 15-E

Recursive Programming

Performance Objectives - Students will be able to:

. Use the concept of recursion in procedures and programs

Vocabulary - New: recursion, IF, STOP
Review: variables

Motivation:

. Copy JKL (Note: This topic has been prepared for Apple Logo.)

```
TO JKL
  FD 54
  REPEAT 18[FD 2 RT 10]
  FD 54 RT 5
END
```

. How would you make five JKLs? 25 JKLs? a thousand JKLs?

- Repeat JKL the number of times desired.
- Use the REPEAT command.

Aim: How can we repeat a procedure without stopping or using REPEAT?

Development:

1. Introduce the concept of recursion.

   . Another way of getting the computer to repeat something, without using REPEAT, is to have a procedure call itself as one of its subprocedures.

2. Copy TRIANGLE:

```
TO TRIANGLE
  REPEAT 3[FD 100 RT 120]
  FD 45 RT 70
END
```

3. Use the Editor to change TRIANGLE into a recursive procedure.

4. Make JKL a recursive procedure.

Application:

1. Write a recursive procedure using:

   - . boxes
   - . circles
   - . your own design

2. Make the following procedures recursive:

```
TO QWE
  PU FD 45 PD
  BK 25 RT 20
  PU BK 12 PD
END

TO ASD
  FD 56 RT 45 FD 15 RT 90
  FD 15 RT 45 FD 56 RT 1
END
```

Summary:

1. How are procedures made recursive?

2. How can the concept of recursion be used to change procedures and programs?

Assignments:

1. Use COUNTDOWN (Handout #1) as a subprocedure in an original program.

2. Write a new procedure that will count up.

Handout #1

```
TO COUNTDOWN :NUMBER
  IF :NUMBER = 0 [STOP]
  PR :NUMBER
  COUNTDOWN :NUMBER - 1
END
```

TOPIC 15 - M

Recursive Programming

Performance Objectives - Students will be able to:

. Use the concept of recursion in procedures and programs

Vocabulary - New: recursion
Review: variables, IF, STOP, PR, TEXTSCREEN, CLEARTEXT

Motivation:

. Copy UIOP (Note: This topic has been prepared for Apple Logo.)

```
TO UIOP
  FD 45
  REPEAT 3[FD 23 RT 120]
  RT 67
END
```

. UIOP makes a flag design.

Figure 1:

. How would you make five UIOPs? 25 UIOPs? a thousand UIOPs?

- Repeat UIOP the number of times desired.

- Use the REPEAT command.

Aim: How can we repeat a procedure without stopping or using REPEAT?

Development:

1. Introduce the concept of recursion.

   . Another way of getting the computer to repeat something, without using REPEAT, is to have a procedure call itself as one of its subprocedures.

2. Copy TRIANGLE:

```
TO TRIANGLE
  REPEAT 3[FD 100 RT 120]
  FD 45 RT 70
END
```

3. Use the Editor to change TRIANGLE into a recursive procedure.

4. Make UIOP a recursive procedure.

Application:

1. Write a recursive procedure using:

. boxes
. circles
. your own design

2. Make the following procedures recursive:

```
TO QWE
  PU FD 45 PD
  BK 25 RT 20
  PU BK 12 PD
END

TO ASD
  FD 56 RT 45 FD 15 RT 90
  FD 15 RT 45 FD 56 RT 1
END
```

3. Copy HI and make it recursive.

```
TO HI
  TEXTSCREEN
  PR[GISELLE SAYS:]
  REPEAT 22[PR[MY BROTHER BRYAN IS #1]]
  WAIT 100
  CLEARTEXT
END
```

4. Write a recursive procedure similar to HI that incorporates the PR command.

Summary:

1. How are procedures made recursive?

2. How can the concept of recursion be used to change procedures and programs?

Assignments:

1. Use COUNTDOWN (Handout #1) as a subprocedure in an original program.

2. Write a new procedure that will count up.

Handout #1

```
TO COUNTDOWN :NUMBER
  IF :NUMBER = 0 [STOP]
  PR :NUMBER
  COUNTDOWN :NUMBER - 1
END
```

TOPIC 15-A

Recursive Programming

Performance Objective - Students will be able to:

. Use the concept of recursion in procedures and programs

Vocabulary - New: Recursion
Review: Variables, IF, STOP, PR, TEXTSCREEN, CLEARTEXT

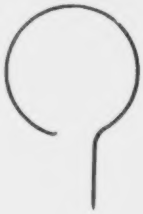
Motivation:

. Copy SHAPE

```
TO SHAPE
  REPEAT 30[FD 10 RT 10] LT 35 FD 50
END
```

. SHAPE makes an incomplete circle with a small line sticking out.

A



. How would you make five SHAPEs? 25 SHAPEs? a thousand SHAPEs?

- Repeat SHAPE the number of times desired.

- Use the REPEAT command.

Aim: How can we get the computer to repeat a procedure without stopping or using REPEAT?

Development:

1. Introduce the concept of recursion. (Note: This topic has been prepared for Apple Logo.)

   . Another way of getting the computer to repeat something, without using REPEAT, is to have a procedure call itself as one of its subprocedures.

2. Copy TRIANGLE:

```
TO TRIANGLE
  REPEAT 3[FD 100 RT 120]
  FD 45 RT 70
END
```

3. Use the Editor to change TRIANGLE into a recursive procedure.

<u>Applications</u>:

1. Write a recursive procedure using:

   . boxes
   . circles
   . your own design

2. Make the following procedures recursive:

```
TO QWE
  PU FD 45 PD
  BK 25 RT 20
  PU BK 12 PD
END

TO ASD
  FD 56 RT 45 FD 15 RT 90
  FD 15 RT 45 FD 56 RT 1
END
```

3. Copy HI and make it recursive.

```
TO HI
  TEXTSCREEN
  PR[GISELLE SAYS:]
  REPEAT 22[PR[MY BROTHER BRYAN IS #1]]
  WAIT 100
  CLEARTEXT
END
```

4. Write a recursive procedure similar to HI that incorporates the PR command.

5. Use BLOCKS or any of its variations as a subprocedure in a program.

```
TO BLOCKS :SIZE
  IF :SIZE < 3 [STOP]
  REPEAT 4[FD :SIZE RT 90]
  FD :SIZE
  BLOCKS :SIZE * .5
END
```

Summary:

1. How are procedures made recursive?

2. How can the concept of recursion be used to change procedures and programs?

Assignments:

1. Use COUNTDOWN (Handout #1) as a subprocedure in an original program.

2. Write a new procedure that will count up.

Handout #1

```
TO COUNTDOWN :NUMBER
  IF :NUMBER = 0 [STOP]
  PR :NUMBER
  COUNTDOWN :NUMBER - 1
END
```

TOPIC 15-E

Using Variables

Performance Objectives - The student will be able to:

- Define procedures that will permit the computer to make the same figures in different sizes.
- Create interesting graphics using variable procedures.

Vocabulary: variable, input

Aim: How can we make procedure that will create the same figures in different sizes?

Development:

1. Commands like FORWARD (FD), BACK (BK), RIGHT (RT), and LEFT (LT) need more input before Logo can act on them.

   - For example, if you type FD and press RETURN, Logo will say, "Not enough inputs to forward." That means you have not indicated how far you want the turtle to walk or turn.
   - These commands require a numerical input.

2. Procedures can use an input also. Here is a typical procedure:

```
TO CIRCLE
  REPEAT 36[FD 8 RT 10]
END
```

   - The FD input determines the size of that circle. In order to change the size, making it either smaller or larger, you have to change the FD input.
   - You can write one procedure that would draw any size circle without editing the procedure. You can do this by using a variable input for FD. The variable procedure for making a circle would be written like this:

```
TO CIRCLE :SIZE
  REPEAT 36[FD :SIZE RT 10]
END
```

   - When the circle has been defined, and you want Logo to make one, type: CIRCLE 10, or CIRCLE 15, or CIRCLE # (where # indicates any number).

3. In a similar manner, you can use a shortcut by teaching the computer a procedure for drawing many different sized squares by using a numerical variable this way:

```
TO SQUARE :SIDE
  REPEAT 4[FD :SIDE RT 90]
END
```

. Now try making these squares: SQUARE 45, SQUARE 31, SQUARE 73

4. A variable procedure for making triangles is this:

```
TO TRIANGLE :S
  REPEAT 3[FD :S RT 120]
END
```

<u>Summary</u>:

1. How can a figure be created in a variety of sizes?

2. How can interesting figures be created using multi-sized figures?

<u>Assignments</u>:

1. Make up three variable procedures of your own.

TOPIC 16-A

Using Variables

Performance Objectives - Students will be able to:

- Use the same procedure to create figures of different sizes, angles, and rates of increase.
- Use variable procedures to produce startling graphic effects.

Vocabulary: input, variable, recursion

Motivation:

Aim: How can we use different inputs to create interesting variable procedures?

Development:

1. Define the following terms:

   Input -- the value that you give to the variable
   Variable -- a word or character that stands for more than one value
   Recursion -- repetition caused by a procedure calling itself

2. If you type this procedure, the turtle draws a square box of a certain size:

```
TO BOX
  REPEAT 4[FD 50 RT 90]
END
```

- This same program can be used to draw square boxes of various sizes if the input for size (i.e. the box's length) can be changed at will.

- You can replace the specific number for the box's length--that is, the FD numerical input, which in this case is 50--with two dots (:) and a name.

  - For example: FD :SIDE, or FD :LENGTH, or FD :S, etc.

- The dots and the variable name must also be included now in the title line of the procedure, like this:

```
TO BOX :SIDE
  REPEAT 4[FD :SIDE RT 90]
END
```

- Note: When you type TO BOX, if your computer responds that BOX is already defined, use the Editor's mode to change your procedure. Type ED "BOX and press RETURN.

- Now try: BOX 30, BOX 10, BOX 70, etc.

2. Change these procedures so that they use a variable size input:

```
TO TRIANGLE
  REPEAT 3[FD 60 RT 120]
END

TO CIRCLE
  REPEAT 36[FD 7 RT 10]
END

TO STAR
  REPEAT 12[FD 50 RT 150]
END
```

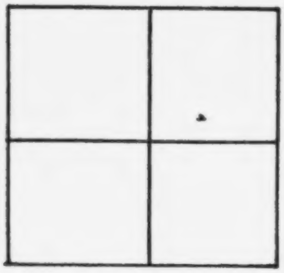
<u>Summary</u>:

1. What is recursion?

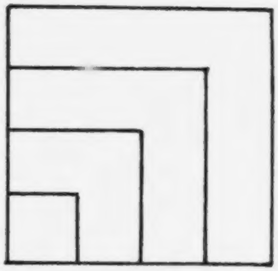
2. How are variables used with recursive procedures?

<u>Assignments</u>:

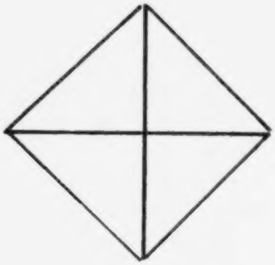
1. Using variable procedures, try drawing these figures:



A



B



C

2. The following procedures have variable inputs and involve recursion. The turtle repeats the procedure until you press CTRL G on the Apple.

```
TO POLY :STEP :ANGLE
  FD :STEP
  RT :ANGLE
  POLY :STEP :ANGLE
END
```

. Now try typing in:

```
POLY 30  60
POLY 40  72
POLY 30  144
POLY 30  160
```

. In the POLY procedure, the turtle draws closed figures. In a spiral the turtle does not end up where it started. It ends up farther away from the starting point each time it goes around.

. In the recursive line of the spiral procedure, you add a little to the step variable in order to get the spiral effect. Type this:

```
TO SPI :STEP :ANGLE
  FD :STEP
  RT :ANGLE
  SPI :STEP +6 :ANGLE
END
```

- Try it:

```
SPI 5 90         SPI 5 125

SPI 5 120        SPI 5 160
```

. You can make STEP variable in SPI by changing the procedure thus:

```
TO SPI :STEP :ANGLE :INC
  FD :STEP
  RT :ANGLE
  SPI :STEP +INC :ANGLE :INC
END
```

- Now try:

```
SPI 5 75 1

SPI 5 75 2
```

TOPIC 17 - E

Using Several Variables

Performance Objectives - Students will be able to:

- Understand the concept of variables.
- Understand the effects of changes in each variable.
- Use programs with several variables to draw diagrams combining the same shape in different sizes.

Vocabulary: variable, container (:side), rectangle diminishing, constant, stack

Motivation:

- How are the directions for drawing a rectangle different from those for drawing a square?
- How many variables do you have to change in order to increase the height of a rectangle?... the width?
- How many variables do you have to change in order to modify the height and the width of a rectangle?
- How can we write a program to draw any sized rectangle?

Aim: How can we write and use rectangle programs that have several variables?

Development:

1. How are the instructions for a variable square program different from the instructions for a rectangle program?
2. How can we write a program to draw any sized rectangle?
3. How can we use the variable rectangle program to draw a stack of adjacent rectangles with constant heights but diminishing widths (diminishing heights but constant widths)?
4. Variable rectangle program:

```
TO RECTANGLE :SIDE1 :SIDE2
  REPEAT 2[FD :SIDE1 RT 90 FD :SIDE2 RT 90]
END
```

Summary:

1. How can we write a program with several variables to explore the properties and uses of rectangles?

Assignments:

1. Use the variable rectangle program to draw a stack of adjacent rectangles with the same height but diminishing widths (diminishing heights but constant widths).

2. Use the variable rectangle program to create your own picture.

3. What other standard shape can be made using the variable rectangle program?

TOPIC 17 - M

Using Several Variables

Performance Objectives - Students will be able to:

- Understand the concept of variables.
- Understand the effects of changes in each variable.
- Write programs with several variables to draw diagrams combining the same shape in different sizes.

Vocabulary: variable, container (:side), circle, angle

Motivation:

- How can we write a program with several variables to draw different sized circles?
- How will changes in each variable affect the size of the circle?
- To what family of shapes is the circle related? How can you demonstrate it?

Aim: How can we write and use circle programs that have several variables?

Development:

1. How can we write a program to draw any sized circle?
2. How does changing the number of repeats affect the size of the circle?
3. How does changing the FD variable affect the size of the circle?
4. How do the instructions for a variable square program and that of circle differ?
5. How does the number of repeats affect the angle of each turn? (Angle = 360/:N)
6. How can we use the variable circle program to draw the following:

   - Variable Circle Program:

```
TO CIRCLE :N :S
  REPEAT :N[FD :S RT 360/:N]
END
```

Summary:

1. How can we write a program with several variables to explore the properties and uses of circles (polygons)?

Assignmments:

1. Write a variable circle program to draw different sized circles that are tangent (touching) at the same point.

2. Use your variable circle program to create your own pictures.

3. Name four polygons you can generate using the variable circle program when you reduce the number of repeats.

4. Describe how you can control the size of the circle.

TOPIC 17-A
Using Several Variables

Performance Objectives - Students will be able to:

- Understand and apply the concept of dependent and independent variables.
- Explore the effects of changes in each variable.
- Write programs with several variables to draw circles of any given radius.
- Investigate the properties of Spi and Inspi programs.

Vocabulary: independent variable, dependent variable, circle, radius, angle

Motivation:

- How can we write a program that generates a circle with any given radius?

- How will changes in each variable affect the shapes generated by the Spi and Inspi programs?

- To what family of shapes is the circle related? How can you demonstrate it?

Aim: What tools can we create to help us investigate the Spi and Inspi programs? (ie. charts, tables, print out of variable values, etc.)

Development:

1. How can we write a program to draw a circle with a radius of any given size?

2. How does the change in the FD (or angle) variable affect the shape of the Spi (Inspi)?

3. How does the length of the radius affects the size of the circle?

4. The programs are:

```
TO CIRCLE :N :R
  REPEAT :N[FD (6.28/:N) * :R RT 360/:N]
END
```

- Explanation:

```
2 II R   = :N * :AMT
2(3.14)R = :N * :AMT
(6.28 / :N) * :R = :AMT
```

- Look at instructions for FD in circle program.

```
TO SPI :A :B :C
  FD :A
  RP :B
  SPI :A + :C :B :C
END


TO INSPI :A :B :C
  FD :A
  RT :B
  INSPI :A :B + :C :C
END
```

Summary:

1. How can we explore the use of several variables through the circle, Spi and Inspi programs?

Assignments:

1. Write a variable circle program using 72 (18, 10) repeats.
2. Can you write a program where both the number of repeats and the radius will be variables?
3. Can you identify the independent and dependent variables and show their relationships? (:AMT = 6.28 / :N * :R, (:Angle = 360 / :N)
4. Compare the instructions for the Spi and Ispi programs.
5. Compare the results of these two programs.
6. Devise tools and investigate what these programs do and why they do it.

TOPIC 18

Hide the Turtle

Performance Objectives - Students will be able to:

- Use the HT command to get a clearer view of how more complicated figures are constructed.
- Make more intricate drawings with a hidden turtle so that its movements can be more accurately planned for.

Vocabulary variable, recursive, HIDETURTLE (HT)

Motivation:

- How can we increase the intricacy of our graphic design by removing the turtle?

Aim: To use variables to increase the intricacy of graphic designs.

Development:

Application:

1. Teach your computer this procedure which takes variable inputs in size and angle:

```
TO POLYSPI :SIZE :ANGLE
   FORWARD :SIZE
   RIGHT :ANGLE
   POLYSPI (:SIZE +1) :ANGLE
END
```

2. Hide the turtle by pressing HT.

   Try these commands. Since the procedure is recursive, you will need to press the BREAK (for Atari) or CTRL-G (for Apple) key when you want to stop the recursion:

```
POLYSPI 1 45        POLYSPI 1 90
POLYSPI 1 88        POLYSPI 1 92
POLYSPI 1 117       POLYSPI 1 120
POLYSPI 1 124
```

3. Experiment by:

   a. Keeping the angle constant and varying the size, for example:

```
POLYSPI 10 100
    POLYSPI 240 100
        POLYSPI 100 100
```

b. Using very small numbers for both inputs, for example:

```
POLYSPI 1 2
     POLYSPI 5 1
```

c. Using normal size input, like 50 or 100, and very small angle input like 1 or 2, for example:

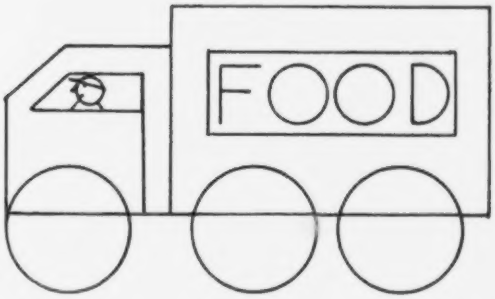
```
POLYSPI 100 1
    POLYSPI 100 3
         POLYSPI 50 2
```

d. Using normal size inputs with huge angle inputs, like 1000 or 20000, for example:

```
POLYSPI 93 1000
    POLYSPI 85 5000
         POLYSPI 98 9999
```

Assignment:

1. Get your computer to draw this picture. Hide the turtle when drawing the inside shapes as much as possible in order to be as accurate as possible.



A

TOPIC 19-A

Squares

Performance Objectives - Students will be able to:

- Edit procedures for making a square in order to make transformations.
- Use the square as a subprocedure in creating more complex programs.
- Use variables and recursions in creating square designs.

Vocabulary: transformation, contraction, dialation, rotation, inversion

Motivation:

- Ask students to illustrate on paper their answer to the following question: What are three ways you can change a square without changing its basic shape at all?
- Have students share the methods they found with the class. (Enlarge, shrink, rotate.)

Aim: How can squares be transformed and rotated? How can variables and recursions be used in complex square programs?

Development:

1. Explain to students how geometry deals with transformations. Explain the following terms to students:
   - A reflective image is the same as a mirror image. The size and shape of the figures is identical.
   - A contraction is a smaller version of the image, which does not change its shape.
   - A dialation is an enlarged version of the original shape.
2. Ask students to create a one-line procedure that will give them a square, a mirror image of that square, a dialation or a contraction. (They will need to write a procedure for square with size as a variable to do this.)

```
TO SQUARE :SIZE
  REPEAT 4 [FD :SIZE RT 90]
END
```

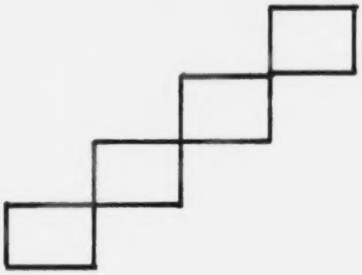
3. Explain that geometry also deals with rotations. Rotation means turning the figure around. If you rotate the figure far enough, you will have an inversion (the figure will be upside down).
4. Write a procedure to rotate a square, so that it ends up as a diamond. Ask the students how far they had to rotate the square (45 degrees).

5. Have students do WORKSHEET 1 (a cube puzzle using rotation).

6. Write programs that use recursions and variables:

   - Have students write a procedure to make a staircase out of boxes, using a recursion. Draw the staircase on the board (see illustration). Elicit from students subprocedures involved. Elicit the following procedures:

```
TO BOX :SIZE
  REPEAT 4 [FD :SIZE RT 90]
END

TO STAIRS :SIZE
  BOX :SIZE
  FD :SIZE
  RT 90
  FD :SIZE
  LT 90
  STAIRS :SIZE
END
```



   - Have students run the program on the computer and note what happens. (The program will look like graph paper when executed.) Elicit from students the reason why. (Because the turtle is in the center and the first box is made to its right.)

7. Have students write a procedure using recursion to make a ladder out of squares.

   - Elicit from students a procedure to make as many boxes as they want in their ladder, using a variable instead of a recursion in their procedure.

   - To make as many boxes as you want, do the following:

```
TO LADDER : X
  REPEAT :X [BOX FD 50]
END
```

8. Have students explore properties of a square by doing PROJECTS 1 and 2.

9. Procedures to make a square grow or become smaller:

   - Elicit from students the procedure to make the box with size as a variable.

```
TO BOX :SIZE
  REPEAT 4[FD :SIZE RT 90]
END
```

- Use a recursion in the rotation procedure that specifies how much larger the square should become with each successive rotation.

  - Example:

```
TO GROW.SQUARE :SIZE
  BOX :SIZE
  RT 17
  GROW.SQUARE :SIZE+5
```

- Write a procedure that makes the box shrink with successive rotations. Ask students to observe what happens. (If you use a number like :SIZE-5 the boxes will get smaller and then bigger because of the negative values on the screen makes it go that far in the opposite direction.)

- Ask students what command they can use to overcome this problem. (The IF THEN command is the non recursive way out.)

- Have students rewrite the program so that the program will stop running before it hits negative values for size.

  - Answer:

```
TO SHRINK :SIZE
  IF :SIZE [ 5 [STOP]
  BOX :SIZE
  RT 17
  SHRINK :SIZE-5
END
```

- Have students think of another way of doing this. (Use a geometric progresion to shrink size.)

```
SHRINK :SIZE = SIZE*.9
```

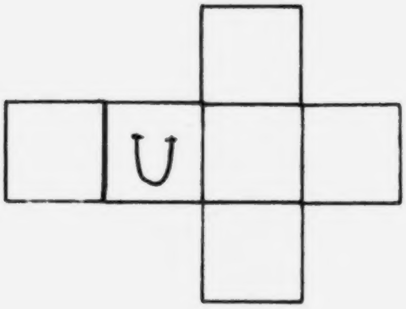
<u>Assignment</u> <u>for</u> <u>More</u> <u>Advanced</u> <u>Students</u>:

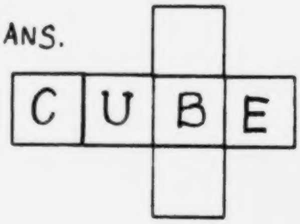
1. THE MELTING ICE CUBE

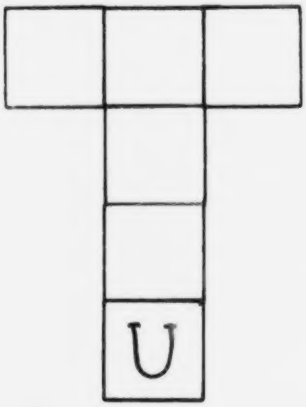
   - Write a program to make an ice cube. Use variables in your procedures. Then write a procedure that erases the larger ice cube, and makes successively smaller and smaller ones, so that it looks like the ice cube is melting.

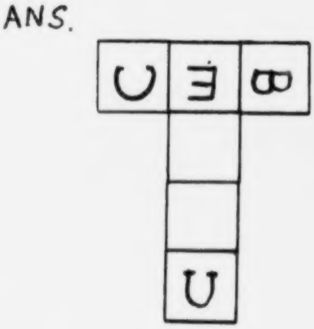
WORKSHEET 1

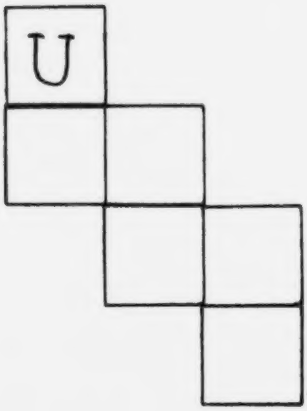
CUT OUT THE FOLLOWING FIGURES. THEN, WITHOUT FOLDING THE FIGURE, WRITE IN THE LETTERS C,B,AND E SO THAT IF THE FIGURE WAS FOLDED INTO A CUBE SHAPE, THE WORD "CUBE" WOULD BE WRITTEN OUT CORRECTLY.

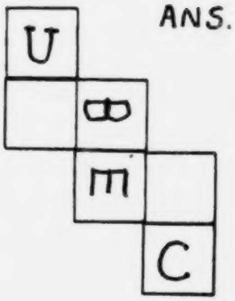












PROJECTS

1. Inscribe the largest possible circle inside a square that touches the square in four places. In order to do this you must find the center of the square, figure out the radius the circle would have, and then the circumference the circle would have.

2. Inscribe the largest square possible inside a circle.

TOPIC 20 - A

Triangles

Performance Objectives - Students will be able to:

- Create procedures for making scalene, isoceles, congruent and similiar triangles.
- Use the triangle as a subprocedure in creating more complex procedures.
- Use variables and recursions in creating triangular designs.

Vocabulary: supplementary, complementary, scalene, isoceles, congruent, similar

Motivation:

- Is every three-straightsided plane figure a triangle? Why?

Aim: How can recursions and variables be used in creating complex rectangular designs?

Development:

1. Elicit from students the definitions of the following angles by drawing them on the board and having students observe their properties.
   - Right angle (90 degrees).
   - Acute angle (< 90 degrees).
   - Obtuse angle (> 90 degrees).
   - Straight angle (180 degrees).
   - Complementary angles (complimentary angles have a sum of 90 degrees).
   - Supplementary angles (supplementary angles have a sum of 180 degrees).
   - Adjacent angles (common ray, and side).
   - Vertical angles (non adjacent).
   - Congruent angles (angles that are the same size).
2. Have students write a procedure to create each of the preceeding angles and to record their procedures in their Logo logs.
   - Ask students to observe the amount the turtle had to turn RT or LT when making an acute angle. Elicit from students why the turtle had to turn the supplemetary amount. (The turtle was originally going FD or BK in a straight line or 180 degrees. It had to turn more than 90 degrees to create an acute supplementary angle.)

3. Elicit the definitions of each of the following triangles. Draw examples of each type on the board and have students examine their properties.

   - Scalene Triangle - a triangle with no equal sides or angles.

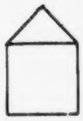
   - Isoceles Triangle - a triangle with two equal sides and two equal angles.

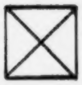
4. Have students write a procedure to create each type of triangle. Then, let the students create designs using these triangles.

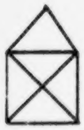
   - Have students do PROJECTS 1 and 2.

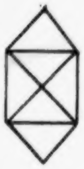
5. Drawing figures without going over lines twice:

   - Put the following figures on the board and label them A, B, C, and D. Ask students to try to draw each of the figures without lifting their pencils off the paper and without going over any line twice. Explain that one of these figures cannot be drawn this way. Ask them to figure out which one it is and what method they used in each of the possible cases.









     - Square with equilateral triangle on top (house shape).

     - Square divided in 4 parts diagonally.

     - Square divided in 4 parts diagonally with an equilateral triangle on top.

     - The same as figure C, with a triangle on the bottom too (Answer B is not possible.)

     - Where did you start and end in A,C and D? (One odd vertex has to be the starting point and one has to be the ending point.)

6. Using this information, ask students to create a Logo program to make a house without going over lines twice.

```
TO HOUSE
  FULLSCREEN
  REPEAT 4 [RT 90 FD 20]

TO TRI
  RT 30
  FD 20
  RT 120
  FD 20
```

   - Make a row of houses that begin at the roof (begin with PU LT 90 PD).

   - Make a row of houses that start on the ground.

- Begin with

```
PU FD 20 PD
HOUSE
PU RT 30 FD 20 RT 180 PD
```

7. Have students do PROJECTS 3-7

PROJECTS

1. Write a program to fill in one of the triangles that you made. Use the variable technique that will make successivly smaller similar triangles. Remember to use the IF THEN statement to avoid negative values.

2. Write a program that rotates a triangle and has it grow or shrink as it is rotated. You may use the IF THEN command to stop the procedure when you have the number of triangles that you want.

3. Make a row of houses using size as a variable.

4. Inscribe the largest possible triangle in a circle.

5. Inscribe the largest possible circle in an equilaterial triangle. Then inscribe the largest circle possible in an isoceles triangle.

6. Make a geodesic dome out of triangles.

7. Make a desert scene with several pyramids of different sizes.

TOPIC 21 - A
Rectangles

Performance Objectives - Students will be able to use:

- Edit procedure for making a rectangle, in order to make other quadrilaterals.
- The rectangle as a subprocedure in creating more complex procedures.
- Variables and recursions in creating rectangular designs.

Vocabulary: quadrilateral, trapazoid, parallelogram, rhombus

Motivation:

- What is the definition of a rectangle? If we changed the length of the sides would we necessarily have a rectangle? (Yes, since a square is a special case a of rectangle, to get a non-square rectangle, opposite sides must be equal, and adjacent sides are not.)

- What other shapes could we get if we only changed length? (None, since only changing length would not affect the angles and any closed figure with four straight sides and four right angles is a rectangle.)

- If we changed the size of the angles could we possibly still end up with a rectangle? (No, 90-degree angles are part of the definition of rectangle.)

- What other shapes could we have if we only changed angle size?

Aim: How can procedures to make a rectangle be altered to make other quadrilaterals? How can rectangles be used as subprocedures in complex programs? How can recursions and variables be used to make complex rectangular designs?

Development:

1. Elicit from students the definition of quadrilateral.

    - Draw several quaderaliterals on the board.

    - Ask students to observe differences and similarities between them and to come up with a definition that is true of all of them.

    - What is a quadrilateral? (any 4-sided figure)

    - Is a rectangle a quadrilateral? (Yes, it is one type of quadrilateral.)

2. Have students examine special quadrilaterals and edit commands to make a rectangle in order to create the other special quadrilaterals. Do not give students the definition of these figures, although you will draw them on the board. Have them come up with the definition by creating a procedure to create the figure, and observing it's characteristics.

   . Draw a trapizoid on the board. (A quadrilateral with 1 pair of opposite sides parrallel, and the other sides not.)

   . Have students edit rectangle procedure to create a trapazoid.

   . Have students measure the angles by adding up the amount the turtle turned RT or LT. Do the angles add up to 360 degrees? Are they all different? Which are the same? Have them also observe comparative size length. When they are finished have them write down their observations as a definition of trapezoid.

3. Draw a parallelogram on the board. (Quadrilateral in which both pairs of opposite sides are parallel.)

   . Have students go through the same process they did with trapazoid and add the definition of parallelogram.

4. Draw a rhombus on the board. (A parallelogram with 4 equal sides.) After students edit the rectangle procedures to create a rhombus, and write down observations about angles and sides, have them add the definition of rhombus to their list.

5. Have students do projects 1 and/or 2.

6. Have students explore properties of the rectangle by doing projects 3 and 4.

7. Writing programs that use rectangles and variables.

   . Have students write a procedure for making a rectangle with size as a variable.

   . How many variables would you need and why? (Two, because rectangles have adjacent sides of different sizes.)

8. Have students use the rectangle with variable size as a subprocedure in a new program. See project 4.

9. Have students use the rectangle with recursion as a subprocedure in a new program. See project 5.

Assignments for More Advanced Students:

1. Write a Logo program to teach other students the definitions of the following quadrilaterals: square, rectangle, trapazoid, parallelogram, and rhombus.

   . Write procedures for each of these shapes and then print information about these quadrilaterals. You may do want to write this program two parts. First show the graphics and explain the characteristics of each quadrilateral. Then, make up a test to see if the other students understand the definitions. Write procedures to read their answers and respond to them.

PROJECTS

1. Write a procedure for a boat. Use a trapezoid for the hull. Make a rectangular mast and triangular sails. Include portholes. Try making it look 3-dimensional!

2. Create your own flag or emblem. Create a design for your flag using a parallelogram, rhombus, trapazoid and rectangle. You may want to draw it on graph paper first.

3. Make a rectangle. Draw a diagonal line from one corner to the opposite corner in order to make two triangles.

4. Edit one of these projects. Use size as a variable in your procedures. Make your project larger or smaller.

5. Write a program, using recursion, to make a city block of apartment buildings. You should use size as a variable in your rectangle procedure, so that your buildings and windows could be different sizes. Use the IF THEN statement so that you do not end up with too many buildings.

TOPIC 22 - A

Circles

Performance Objectives - Students will be able to:

- Make a circle.
- Define procedures that make circles.
- Use circles as subprocedures in programs.
- Define procedures that make circles using a variable for the angle.

Vocabulary - Review: REPEAT, : ("dots")

Motivation:

- Copy the following: FD 8 RT 10. (Note: This topic has been prepared for Apple Logo.) Copy it again and again. Keep repeating the same commands. After FD 8 RT 10 is repeated various times:
- What do you think the turtle is drawing?
- How many times do you think you have to repeat FD 8 RT 10 to complete the circle?
- What will happen if we change FD 8 to FD 3 and we repeat these commands?

Aim: How can we use circles to enhance procedures?

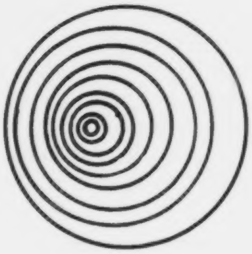
Development:

1. Copy:

```
    REPEAT 36[FD 1 RT 10]
    REPEAT 36[FD 2 RT 10]
    REPEAT 36[FD 3 RT 10]
    REPEAT 36[FD 4 RT 10]
    REPEAT 36[FD 5 RT 10]
    REPEAT 36[FD 6 RT 10]
    REPEAT 36[FD 7 RT 10]
    REPEAT 36[FD 8 RT 10]
    REPEAT 36[FD 9 RT 10]
    REPEAT 36[FD 10 RT 10]
  END
```

2. Write a procedure that will make the design made above. Call it CIRCLES.

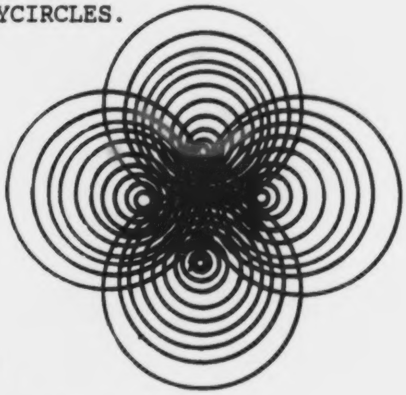
A



3. Copy: REPEAT 4[CIRCLES RT 90]

4. Write a procedure called MANYCIRCLES.

A



5. Copy: REPEAT 360[FD 1 RT 1]

6. How can this circle be made smaller?

   . Use inputs less than one. For example:

     - REPEAT 360[FD .5 RT 1]

     - REPEAT 360[FD .07 RT 1]

7. Copy:

```
TO C :SIZE
  REPEAT 36[FD :SIZE RT 10]
END
```

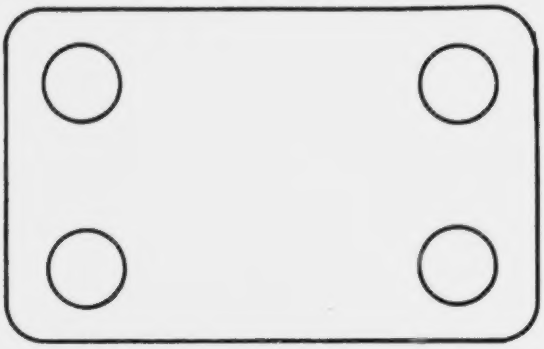
8. Using C, write two procedures that make drawings found in Figure A and in Figure B.

Applications:

1. Write a procedure that will put a circle in each of the four corners of the video monitor.

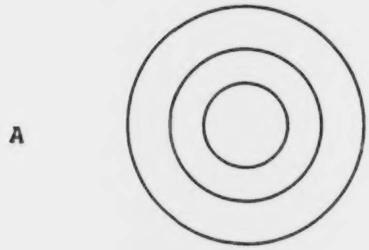
```
TO CIRCLE
  REPEAT 36[FD 4 RT 10]
END
```

B



2. Write a program that uses CIRCLE as a subprocedure.

3. Make a "bull's eye."



4. Write a procedure that will make a "bull's eye."

5. Edit the "bull's eye" procedure. Use SETPC to make the rings with different colors.

6. Copy CIRCLE1 and CIRCLE2.

```
TO CIRCLE1 :SIZE
  REPEAT 360[FD :SIZE RT 1]
END

TO CIRCLE2 :RADIUS
  CIRCLE1 :RADIUS * .0174
END
```

7. Use CIRCLE2 and CIRCLE1 as subprocedures in a program.

<u>Summary</u>:

1. Copy the following three procedures:

```
TO X
  REPEAT 36[FD 5 RT 9]
END

TO Y
  REPEAT 36[FD 5 LT 10]
END

TO Z
  REPEAT 30[FD 5 RT 10]
END
```

2. Which procedure makes a complete circle?

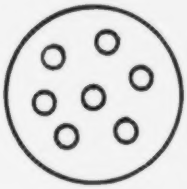
3. Which procedure makes an incomplete circle?

4. Change X, Y, and Z so that each angle in each procedure has a variable.

Assignments:

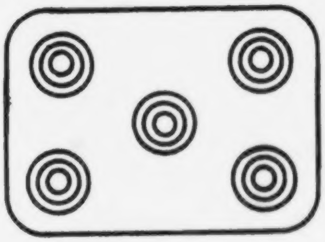
1. Write a program that makes a face using circles of different sizes as subprocedures.

2. Write a program that puts small circles inside a larger circle.



A

3. Write a program that puts four "bull's eyes" in the corners of the monitor screen and one in the center.



B

TOPIC 23 - A

Colors

Performance Objectives - Students will be able to:

- Change the screen's background color.
- Change the turtle's pen color.
- Write procedures that integrate background and pen colors within other procedures.

Vocabulary - New: SETBG, SETPC
Review: IF, STOP, SETPOS, FULLSCREEN

Motivation:

- Boxes (Handout #1) demonstrates four colors (green, violet, orange, blue).

  - Note: This topic has been prepared for Apple Logo.

- How can you make the green box appear in the lower left rather than the top right without changing the procedure?

Figure 1:

Blue	Green
Orange	Violet

Figure 2:

Violet	Orange
Green	Blue

- If you change green and orange, will the blue and violet also change?

- Can one color be replaced without moving the others? (Only by changing the procedure.)

- How can BOXES be changed so that it also demonstrates four or more of the available background colors?

Aim: How can we use color changes to change and enhance procedures?

Development:

1. Copy SUN and SUNS (handout #2).

2. How can SUNS be changed so that each figure is made with a different color?

3. Copy TREE (handout #3).

4. Use SETPC to add color to the balls on the tree. Do not use SETPC 0 (black) or SETPC 2 (green).

Applications:

1. Write a new program that uses SETBG and SETPC in any of its subprocedures.
2. Add SETBG and SETPC to a previously defined procedure.
3. Add more colored balls to TREE.
4. Change SUNS so that you have a geometric design with seven figures instead of five.

Summary:

1. How can SETBG and SETPC be used to enhance procedures?

Assignment:

1. Using BOXES as a guide, write a similar procedure using circles. (The procedure should stop at :SIZE > 10.)

Handout #1:

```
TO BOX :SIZE
  IF :SIZE = 50 [STOP]
  REPEAT 4[FD :SIZE RT 90]
  BOX :SIZE + 1
END
```

```
TO BOXES
  SETPC 2 BOX 0 RT 90
  SETPC 3 BOX 0 RT 90
  SETPC 4 BOX 0 RT 90
  SETPC 5 BOX 0 RT 90
END
```

Handout #2:

```
TO SUN
  REPEAT 40[FD 30 BK 28 RT 9]
END
```

```
TO SUNS
  FULLSCREEN
  PU SETPOS [75 75] PD SUN
  PU SETPOS [-75 -75] PD SUN
  PU SETPOS [-75 75] PD SUN
  PU SETPOS [75 -75] PD SUN
END
```

Handout #3:

```
TO LEFTSIDE
  SETPC 2
  PU RT 90 FD 10 LT 90 BK 25 PD
  BK 34 FD 34 LT 90 FD 55 RT 135
  REPEAT 6[FD 30 RT 25 BK 15 LT 25]
  FD 30 RT 90
END
```

```
TO RIGHTSIDE
  SETPC 2
  REPEAT 6[FD 30 LT 25 BK 15 RT 25]
  FD 30 LT 45 BK 55 LT 90
  BK 34 RT 90 BK 19
END
```

```
TO BALL
  BB 0 LT 40
  REPEAT 36[FD 1 RT 10]
END
```

```
TO BB :P
  IF :P > 7 [STOP]
  REPEAT 4[FD :P RT 90]
  BB :P + 1
END
```

```
TO BALLS
  PU LT 90 FD 57 PD BALL PU
  RT 45 FD 24 PD BALL PU LT 78
  FD 13 PD BALL PU LT 90 FD 35
  PD BALL
END
```

```
TO MOVE
  PU SETH 0 FD 105
  RT 90 BK 23 PD
END
```

```
TO STAR
  SETPC 5 REPEAT 5[FD 25 RT 144]
END
```

```
TO TREE
  CS FULLSCREEN
  LEFTSIDE
  RIGHTSIDE
  BALLS
  MOVE
  STAR
END
```

TOPIC 24 - A
Polygons

Performance Objectives - Students will be able to:

- Write a program to generate any regular polygon.
- Write a program to generate variable sized polygons.
- Write a program to generate solid polygons.

Vocabulary: polygon, independent variable, dependent variable, function, angle, recursion

Motivation:

- How many polygons can you name?

- What is the smallest number of sides needed to make a polygon?

- How would you write programs to draw several types of polygons?

- How would you write a program to generate all polygons?

Aim: How can we generate and investigate polygons using Logo?

Development:

1. List the different types of polygons.

2. List the programs for several polygons.

3. Develop an algorithm for polygons.

4. Write a program to generate any polygon.

5. The programs are:

```
TO TRI :SIDE
 REPEAT 3[FD :SIDE RT 90]
END

TO POLY :N :S
 REPEAT :N [FD :S RT 360/:N]
END

TO FILL.POLY :N :S
 IF :S < 0 [STOP]
 POLY :N :S
 FILL.POLY :N :S-1
END
```

Summary:

1. How can we write a program to generate regular polygons?

Assignments:

1. Write a program to generate polygons with variable sides.
2. Write a program to draw solid polygons.
3. Write a polygon program to generate a circle.
4. Write a polygon program to generate different sized circles.
5. Write a program to generate a circle with a given radius.

TOPIC 25-E

Programs in Logo

Performance Objectives - Students will be able to:

- Write commands instructing the turtle to draw simple shapes.
- Re-write the commands using a program format.
- Differentiate between learning to do something and doing it (i.e. to square vs. square).
- Appreciate the importance of the use of END in a program.
- Explore appropriate options for modifying or replacing a program (rename or erase program).

Vocabulary: learn vs. do, command, program, END, rename, path, square

Motivation:

- Can you be the turtle and follow instructions?
- Can the class give the student-turtle instructions to move in a square path?
- Can we list the instructions as a script (program) to follow.
- Can we give the LOGO turtle commands to draw a square?
- Can we list these commands in a program to teach the turtle to draw a square?

Aim: How can we use LOGO to provide students with a script and the turtle with a program to perform tasks?

Development:

1. Have students play the part of the turtle and follow commands given by the class.
2. Write the instructions in a script for the students to perform.
3. Have a student follow the script to draw a square.
4. Write a LOGO program for the turtle to draw a square.
5. Consider the issues of renaming or erasing a program when necessary.

6. Script (program):

```
TO MOVE
  STAND UP
  FACE BLACKBOARD
  WALK FORWARD 30 STEPS
  TURN AROUND
  SMILE
  WALK BACK TO SEAT
  SIT DOWN
END
TO SQUARE
  FD 40 RT 90
  FD 40 RT 90
  FD 40 RT 90
  FD 40 RT 90
END
```

Summary:

1. How can we use LOGO to write scripts for people and programs for the computer to perform given tasks?

Assignments:

1. List instructions for entering (leaving) classroom.
2. List instructions for playing Hide-N-Seek (or other game).
3. Write a script (program) for walking along a small (large) square (other shapes).
4. Write a script (program) for walking along two adjacent squares.
5. Write a program for the turtle to draw a square, two adjacent squares, four squares that form a square.

TOPIC 25 - M

Programs in Logo

Performance Objectives - Students will be able to:

. Use the edit mode to write and modify programs.
. Differentiate between learning to do something and doing it (i.e. to square vs. square).
. Understand the importance of the use of END in a program.
. Create modules to be used in more complex programs.

Vocabulary: learn vs. do, command, program, END, edit mode, modules

Motivation:

. How can you make changes in a program without renaming or erasing it?

. How can you use the edit mode to write a program that draws a square (triangle)?

. How can you use the edit mode to write a program that uses your square and triangle programs in order to draw a house?

. Can you create other modules for your house program (door, windows, etc.)?

Aim: How can we use the edit mode to write and modify programs, and develop modules?

Development:

1. Have students use the edit mode to correct errors (make changes) in simple programs.

2. Have students use the edit mode to develop modules for a house (square, triangle programs).

3. Have students use the modules listed above in a house program.

Summary:

1. How can we use the edit mode to write and modify programs and to develop modules for more complex programs.

Assignments:

1. Use the edit mode to correct the following programs:

```
TO SQUARE
  REPEAT [FD 40RT 90]
END

TO TRIANGLE
  REPEAT 3[FD 40 RT 120 END
```

2. Use a square and a triangle as modules to write a program for a house.

3. Develop additional modules for your house (ie. door, window, etc).

TOPIC 25-A

Programs in Logo

<u>Performance</u> <u>Objectives</u> - Students will be able to:

- Use the edit mode to write and modify programs.
- Use the edit mode to use parts of existing programs to create new ones (while retaining the old ones).
- Use the edit mode to work on several programs at the same time.
- Create modules to be used in more complex programs.

<u>Vocabulary</u>: edit mode, modules

<u>Motivation</u>:

- How can you use the edit mode to write and modify programs?

- How can you use the edit mode to create a program for a square.L from a square.R program? (Rec.L, Circ.L. etc.)

- How can you use the edit mode to work on several modules simultaneously?

<u>Aim</u>: How can we use the edit mode to write, modify, and combine elements of one or more programs?

<u>Development</u>:

1. Use the edit mode to develop modules for a project (house, truck, person, robot, etc.).

2. Have students develop new modules from existing ones where appropriate.

3. Have students use the edit mode to work on several modules simultaneously.

```
TO SQUARE :SIDE
  REPEAT 4[FD :SIDE RT 90]
END

TO TRI :SIDE
  REPEAT 3[FD :SIDE RT 120]
END

TO HOUSE :SIDE
  SQUARE :SIDE
  FD :SIDE
  RT 30
  TRI :SIDE
  PU SETH 0 BK :SIDE PD
END
```

Summary:

1. How can we use the edit mode to write, modify and combine elements of one or more programs?

Assignments:

1. Use the edit mode to create modules for a house (truck, person, robot, etc.).
2. Use the edit mode to develop additional modules from existing ones where appropriate.
3. Use the edit mode to combine above mentioned modules to complete your project.

TOPIC 26 - A

Coordinates

Performance Objective - Students will be able to:

. Use the SETX, SETY, and SETPOS commands in procedures and programs.

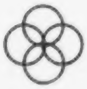
Vocabulary - New: SETX, SETY, SETPOS
Review: PU, PD, FULLSCREEN, HOME

Motivation:

. Copy CIRCLE and RINGS. (Note: This topic has been prepared for Apple Logo.)

```
TO CIRCLE
  REPEAT 36[FD 2 RT 10]
END

TO RINGS
  REPEAT 4[CIRCLE RT 90]
END
```

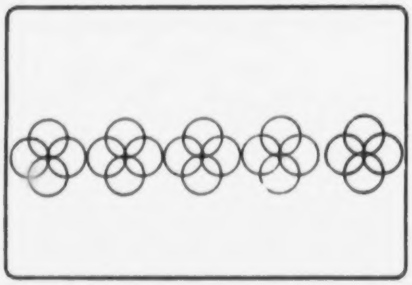
A (RINGS):



. Allow pupils to experiment with RINGS.

. On the board draw a screen with five RINGS in a row.

B



. Using RINGS, what is one way of making this design (Fig. B)?

. After allowing the pupils to demonstrate their solutions, introduce the concepts found in the Cartesian X an Y coordinate system.

- Demonstrate one way of making a series of five RINGS using the the new commands:

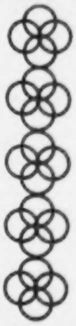
```
CS
RINGS PU SETX 45 PD RINGS
PU SETX 90 PD RINGS
PU SETX -45 PD RINGS
PU SETX -90 PD RINGS
```

Aim: How can we move the turtle to a specific location on the screen without using FD or BK?

Development:

1. Using RINGS, make a series of five vertical rings. Use the same inputs used with SETX to make the five horizontal rings.

A

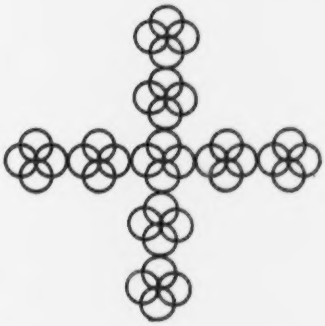


2. Introduce SETPOS. Using SETPOS, how can the previous exercises be solved?

3. Write a procedure that makes the five verticle rings.

4. Write a procedure that draws nine rings in the shape of a "plus" sign.

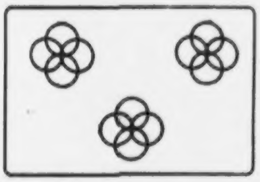
B



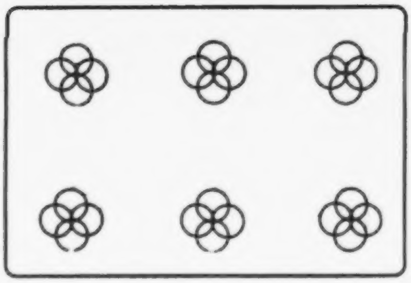
Applications:

1. Using the concepts introduced, write three procedures that make each of the following:

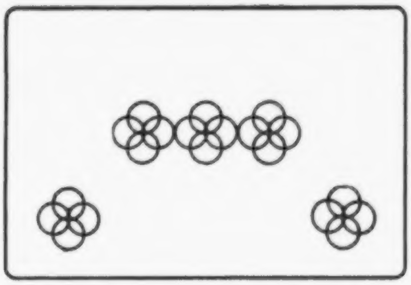
C



A



B



2. Write a program that uses SETX, SETY, and SETPOS in any of its subprocedures.

3. Copy MOVE (Handout #1) and NUMBER (Handout #2).

4. Change MOVE so that it stops after the turtle has drawn 20 lines.

5. Change NUMBER so that the turtle draws in the negative coordinates. (In line 3 of NUMBER, change :NUM to -:NUM.)

Summary:

1. How may we use SETX and SETY to shorten our programs?

2. How may we use SETX and SETY to place the turtle in different spots of the screen?

3. How can SETPOS be used to shorten programs?

Assignments:

1. Write a program that places small drawings in the following places on the screen:

   . SETX 34 SETY 89

   . SETY 12 SETX -10

   . SETPOS [-34 -123]

   . SETPOS [1 100]

Handout #1:

```
TO MOVE
  CLEARTEXT
  SETX NUMBER
  SETY NUMBER
  PR POS
  WAIT 200
  MOVE
END
```

Handout #2:

```
TO NUMBER
  MAKE "NUM RANDOM 100
  IF 1 = RANDOM 2 [OUTPUT :NUM]
  OUTPUT :NUM
END
```

TOPIC 27 - A

Logo and Words

<u>Performance Objectives</u> - Students will be able to:

- Write procedures using the Logo primitive ITEM.
- Write procedures using simple recursion and a single variable in ITEM to print out text messages.

<u>Vocabulary</u> - New: ITEM.
Review: Ordinals (first, second, third, etc).

<u>Motivation</u>:

- We need to be able to designate and change or exchange a particular item in a long list more conveniently than by the excessive use of FIRST LAST BUTFIRST (BF) and BUTLAST (BL).

<u>Aim</u>: How can we use our knowledge of recursion and variables in procedures which will print out messages on the monitor screen?

<u>Development</u>:

1. The Logo primitive ITEM with one input designates a particular item in ordinal sequence (first, second, third, etc.). ITEM may be used with Logo words or lists. (Note: This lesson was written for Apple Terrapin Logo.)

```
PR ITEM 1 [RED YELLOW GREEN]
PR ITEM 7 "CALIFORNIA
PR ITEM 3 [ SIX 3 NINE 3 THREE]
PR ITEM 2 [SUN MON TUE WED THUR]
```

2. We can use ITEM and one input to manipulate different parts of a Logo sentence.

   - Predict the print-out for each of the commands below.

   - Note correct placement of all abbreviations, primitives, and punctuation.

```
PR SE [I LIKE] ITEM 3 [LOLLIPOPS. SOURBALLS. BONBONS. JELLYBEANS.]

PR LPUT ITEM 1 [LOLLIPOPS SOURBALLS CARAMELS] [BONBONS GUMDROPS TAFFY]

PR (SE ITEM 2 [CHERRY GRAPE LEMON] [IS MY FAVORITE] BF [SUMMERTIME
   FRUIT FLAVOR.] )
```

   - Explain each printout above.

3. The Logo primitive ITEM with one input can be used in Logo procedures.

   . Predict the printout for the procedure below.

```
TO COLORS
  PR (SE ITEM 1 [RED ORANGE YELLOW] [IS MY FAVORITE COLOR.] )
  PR[]
  PR (SE LPUT ITEM 1 [BLUE? ORANGE? RED?] [IS YOUR FAVORITE COLOR] )
  PR[]
  PR (SE FPUT ITEM 4 [RED ORANGE YELLOW BLUE GRAY] [IS THE COLOR OF
   THE SKY ON A SUMMER DAY.]
END
```

<u>Applications</u>:

1. Use the input for ITEM as a variable in recursive procedures that require a count. Type:

```
TO SPELLDOWN :COUNT
  PR []
  IF :COUNT = 11 STOP
  PR ITEM :COUNT "CALIFORNIA
  SPELLDOWN :COUNT + 1
END
```

   . Call this procedure by typing: SPELLDOWN 1

   . Type:

```
TO BLAST :COUNT
  IF :COUNT = 6 PR [BLAST OFF!!] STOP
  PR ITEM :COUNT [FIVE! FOUR! THREE! TWO! ONE! ZERO!]
  PR []
  BLAST :COUNT + 1
END
```

   . Call this procedure by typing: BLAST 1

   . Type:

```
TO RAVE :COUNT
  PR []
  PR ( SE [I LIKE TO WORK WITH COMPUTERS ON] ITEM :COUNT [MON. TUES.
     WED. THURS. FRI. SAT. SUN.] "HURRAY!! )
  IF :COUNT  = 7 STOP
  RAVE :COUNT + 1
END
```

   . Call this procedure by typing: RAVE 1

Summary:

1. How have we used ITEM to create interesting Logo procedures?

Assignments:

1. Edit the SPELLDOWN procedure to print out the letters in your first name. Skip a line between each letter.
2. Edit the BLAST procedure to print out numerals instead of the names of the numbers.

Assignments for Advanced Students:

1. Edit the SPELLDOWN procedure to print down the letters in your whole name. Skip lines only between your first, middle, and last names.
2. Edit the RAVE procedure to include the names of your favorite TV shows.

TOPIC 28 - M

Writing with Logo

Performance Objectives - Students will be able to:

- Edit a set of formula procedures which may be used to generate ideas for independent creative writing activities.

Vocabulary - New: OPERATION RANDOM, OUTPUT (OP)
Review: ITEM, subprocedure

Motivation:

- Writers need good sources from which to draw ideas for their stories. Many writers even use the computer to help them with this. How does a writer use a computer to find ideas?

Aim: How can we use Logo to give us ideas for our writing projects?

Development:

1. OUTPUT (OP) is a Logo operation and primitive which can only be used in a procedure. It cannot be used at top level. A LOGO operation gives a value to an object or thing.
(Note: This lesson was written for Apple Terrapin Logo.)

- OUTPUT (OP) produces Logo information,

```
TO DEMO1
  OUTPUT 7 + 4
END

TO DEMO2
  OP ITEM 2 "CAT
END
```

- Type and run:

```
TO SHOW
  PR DEMO1  DEMO2
END
```

2. OUTPUT (OP) can be used to find information in Logo lists.

Type:

```
OP ITEM 5 [RED BLUE YELLOW GREEN ORANGE BLACK WHITE GRAY]
```

3. RANDOM is another LOGO operation and primitive. It can be used to determine the input for ITEM. ITEM can never be zero (0); therefore, 1 should always be added to the randomly generated ITEM input. Note placement of parentheses. Type:

```
OP ITEM (1 + RANDOM 5 ) [RED BLUE GREEN YELLOW ORANGE]

OP ITEM (1 + RANDOM 10 ) "CALIFORNIA
```

<u>Applications</u>:

1. Use the procedures below to print out ideas for use in your creative writing. Notice placement of primitives, abbreviations, and punctuation.

```
TO PICK.NAME
  OP ITEM ( 1 + RANDOM 10 ) [MARTIAN ROBOT STUDENT MR.T MAYOR
     SUPERMAN PRESIDENT PINK.PANTHER MILLIONAIRE TEACHER]
END


TO PICK.VERB
  OP ITEM (1 + RANDOM 10 )[SKIPS FLIES HOPS HIKES WALKS CRAWLS CLIMBS
     ESCAPES GALLOPS RUNS]
END


TO PICK.PLACE
  OP ITEM (1 + RANDOM 10 )[PARIS THE.MOON MARS HOLLYWOOD HAWAII
     PIKE'S.PEAK HOBOKEN PARK.AVENUE BROADWAY SOUTH.POLE]
END


TO PICK.ACT
  OP ITEM (1 + RANDOM 10 ) [FIND EAT GRAB READ SELL STUDY THROW.AWAY
     HUG SHAKE BUG]
END


TO PICK.THINGS
  OP ITEM (1 + RANDOM 10 ) [DINOSAURS COOKIES KANGAROOS DUCKS DAISIES
     LOLLIPOPS BROCOLLI PETUNIAS TEDDY.BEARS GUMDROPS]
END


TO IDEA
  PR (SE PICK.NAME PICK.VERB "TO PICK.PLACE "TO PICK.ACT PICK.THINGS)
END


TO PICK
  REPEAT 5 [IDEA PR[]]
END
```

- Call this program by typing PICK.
- Logo should print out five ideas on your monitor screen for you to use in your next writing project. If necessary, you can run PICK again and again until Logo prints out an idea you can use.

2. Edit each of the procedures above to include names, places, etc. which you prefer.

Summary:

1. How did we use Logo to find ideas for our creative writing projects?

Assignments:

1. Choose one idea from among those printed out by Logo. Use that idea to develop a short story.

Assignments for Advanced Students:

1. Create an original story in which the main idea of each paragraph in the story is based on a different idea as suggested by Logo.

TOPIC 28 - A

Writing with Logo

Performance Objectives - Students will be able to:

. Write procedures to print data reports.

Vocabulary - New: data, database.
Review: report, record, information.

Motivation:

. While some of the writing that we do is creative, such as stories, letters, or poems, we also use our writing skills to make reports or keep records. How can procedures help us with these kinds of activities?

Aim: How do we write Logo procedures to enable us to keep a record of information?

Development:

1. Collect and organize the information that you want to include in your database. For example, you might want to poll your friends for their phone numbers and favorite party food. For example:

Mary	111-1111	Hot.Dogs
John	222-2222	Potato.Chips
Bill	333-3333	Fruit.Punch
Jane	444-4444	Ice.Cream

2. Write a procedure to add information as needed to a standard database. (Note: This lesson was written for Apple Terrapin Logo.) Type:

```
TO MARY
  PR[MARY JONES]
  PR[PHONE: 111-1111 (CALL AFTER 6 PM)]
  PR[FAVORITE PARTY FOOD IS HOT DOGS.]
END

TO JOHN
  PR[JOHN SMITH]
  PR[PHONE: 222-2222 (DON'T LEAVE MESSAGES WITH LITTLE BROTHER)]
  PR[FAVORITE PARTY FOOD IS POTATO CHIPS.]
END
```

Applications:

1. Write a similar procedure for each friend you want to include in this list.

   . Write a procedure to print the list with all the names and pertinent information. Type:

```
TO PARTY.GUESTS
  MARY PR[]
  JOHN PR[]
  BILL PR[]
  JANE PR[]
END
```

   . This procedure skips a line between each subprocedure; this helps to make the printout easier to read and understand.)

Summary:

1. How have we used Logo to print out sets of collected information?

Assignments:

1. Write procedures to record and print a record of the temperatures for each day of a selected week.

Assignments for Advanced Students:

1. Write procedures to record the opinions of your friends about selected TV shows, pop stars, or hit songs.

TOPIC 29 - M

Naming

Performance Objectives - Students will be able to:

- Write procedures using MAKE to create Logo lists.

Vocabulary - New: MAKE.
Review: list, WORD.

Motivation:

- We need to be able to reduce the amount of typing needed to access lengthy Logo lists. How can we use Logo to group these lists into categories with appropriate names?

Aim: How do we write Logo procedures using lists in a shorter, easier form?

Development:

1. It is good practice to group items into appropriate categories for easier handling in speaking and writing.

   - For example, it is very convenient to be able to say "I like flowers", instead of "I like roses, tulips, daffodils, petunias, daisies..." and on and on, until you have named all the flowers you like.

2. The MAKE command groups a number of items together in a list. This does away with the need to type out the entire list whenever we want to use it. (Note: This lesson was written for Apple Terrapin Logo.)

```
PR [ I LIKE TO EAT APPLES PEARS ORANGES PLUMS.]

MAKE "FRUIT [APPLES PEARS ORANGES PLUMS]

PR ( SE [I LIKE TO EAT ] :FRUIT)
```

Applications:

1. We can use the MAKE command to save typing on reports that require that the same line be used in several different places.

```
TO QUICK.TYPE
  MAKE "HERE [PUT YOUR ANSWER ON THIS LINE AT THE RIGHT___________.]
  PR [WHAT IS THE CAPITAL CITY OF NEW YORK STATE?]
  PR :HERE
  PR [ON WHAT CONTINENT IS THE COUNTRY OF FRANCE LOCATED?]
  PR :HERE
  PR [HOW MUCH IS 2 + 2?]
  PR :HERE
END
```

Summary:

1. How do we use MAKE in a Logo procedure to group lists and sentences into manageable categories?

Assignments:

1. Write a procedure in which MAKE is used to group the different kinds of foods that are needed in a good daily diet.

Assignments for Advanced Students:

1. You own a pet store. Write a procedure which combines several subprocedures into a program to print out the types of animals and the foods they eat.

TOPIC 29-A

Naming

Performance Objectives - Students will be able to:

- Write Logo graphics procedures in which a variable is named with the MAKE command.

Vocabulary - New: PO, NAMES
Review: variable, graphics, x- and y- coordinates

Motivation:

- The use of the MAKE command helps us to expand our procedures easily to include more ideas and to print out more information. How can we use the MAKE primitive in turtle graphics procedures?

Aim: How can we use MAKE in variations of the spiral Logo procedures?

Development:

1. Most spiral procedures include at least one variable; here is one with three. (Note: This lesson was written for Apple Terrapin Logo.)

```
TO SPI :SIDE :ANGLE :TIMES
  HT FULLSCREEN
  IF :TIMES = 0 STOP
  REPEAT :TIMES [FD :SIDE RT :ANGLE]
  SPI :SIDE + 2 :ANGLE :TIMES - 100
END
```

- Call this procedure by typing: SPI 5 25 25

2. Since we can change the name of the variable from within the procedure, it is not necessary to type the variable when the procedure is called.

- Edit the procedure above to change the :SIDE variable from within the procedure. Use the MAKE command.

```
TO SPI :ANGLE :TIMES
  DRAW HT FULLSCREEN
  MAKE "SIDE 5
  REPEAT :TIMES [FD :SIDE RT :ANGLE MAKE "SIDE :SIDE +2]
  SPLITSCREEN
END
```

- Call this procedure by typing: SPI 100 100

Applications:

1. **MAKE** can be used in other types of graphics. In the following program it is used in a subprocedure.

```
TO SQUARE :SIDE
  REPEAT 4[FD :SIDE RT 90]
END

TO CORNER
  MAKE "SIDE 5
  SQUARE :SIDE
  MAKE "SIDE :SIDE + 30
  SQUARE :SIDE
END

TO SPIN.SQU
  REPEAT 18[CORNER RT 100 PU FD 5 PD]
END
```

   . Call this procedure by typing: SPIN.SQU

2. In the following subprocedure, **MAKE** is used to rename the coordinates of a randomly selected point on the Logo grid.

```
TO SPOT
  PU RT RANDOM 360 FD RANDOM 50
  MAKE "EX XCOR
  MAKE "WY YCOR
END
```

   . The coordinate names of the starting point are placed in another subprocedure which can be used to reall it for later use.

```
TO POINT
  SET :EX :WY
END
```

   . When lines are drawn from this starting point, the names of its coordinates can be used to instruct Logo where to end in order to make the drawing a closed figure.

3. In some respects, this renaming of coordinates can act similarly to the HOME Logo primitive command.

   . To illustrate this, we will next define ARCK1, a subprocedure which draws arcs of varying sizes.

```
TO ARCK1
  REPEAT 7 [FD 5 RANDOM 15 RT 25]
END
```

4. Now we will use all of these subprocedures to define LUNA, which draws a small half moon-shaped figure. In LUNA, Logo is instructed to:

   - Select a random point.

   - Name the coordinates of that point (EX and WY).

   - Draw an arc starting from those coordinates.

   - Return to the starting point by calling the names of its coordinates.

```
TO LUNA
  DRAW FULLSCREEN HT
  SPOT
  ARCK1
  POINT
END
```

5. To demonstrate how this works with a more complicated figure, we will next edit both the ARCK1 and the LUNA subprocedures to include instructions to Logo to draw four curved lines of random length. These will form four legs of an irregularly-shaped 5-pointed star.

```
TO ARCK
  REPEAT 4 [FD 20 + RANDOM 4 RT 4]
END

TO STAR
  SPOT
  PD REPEAT 4 [ ARCK RT 144]
END
```

   - Finally, by including the POINT subprocedure, we instruct Logo to complete the figure by recalling thge starting point and drawing a straight line directly to it.

```
TO STAR
  DRAW FULLSCREEN HT
  SPOT
  PD REPEAT 4 [ARCK RT 144]
  POINT
END
```

6. The actual values of the coordinates can be learned by instructing Logo to print out the names used in the procedure with the PO command. For example:

```
PO NAMES
```

. This command can also be included in the procedure itself. Include also a SPLITSCREEN command to provide an area for display of the information. Edit the STAR procedure by typing:

```
TO STAR.N
  DRAW CLEARTEXT SPILTSCREEN HT
  SPOT
  PO NAMES
  PD REPEAT 4 [ ARCK RT 144 ]
  POINT
END
```

7. The following procedure uses STAR.N as a subprocedure to draw and wipe ten of the closed 5-pointed figures. As each figure is drawn the names and respective values for the coordinates of the start/end point are printed out at the bottom of the screen.

```
TO STAR10.N
  REPEAT 10 [ STAR.N ]
END
```

<u>Summary</u>:

1. How do we use MAKE to change a variable from within a Logo graphics procedure?

<u>Assignments</u>:

1. Write a procedure which draws a triangular spiral. Use the MAKE command to change one of the variables from within the procedure.

2. Write a procedure which draws a triangle with unequal sides. Draw the third leg of the triangle by using renamed coordinates of the starting point.

<u>Assignments for Advanced Students</u>:

1. Write a procedure which draws a pentagonal or hexangonal spiral. Use the MAKE command to change one of the variables from within the procedure.

2. Write a procedure which draws a house with a pointed roof. Use the MAKE command to rename the coordinates of the highest point on the roof. Use the name of this point to instruct Logo to draw a flag there.

TOPIC 30 - M

Logo Poems

Performance Objectives - Students will be able to:

. Write Logo procedures to print word groupings in patterns resembling blank verse.

Vocabulary - New: Blank verse, PRINT1.
Review: ITEM, OUTPUT (OP), noun, verb, adjective, gerund, and other parts of speech.

Motivation:

. Some well-written poetry appeals to us because interesting words are used in pleasing rhythmic patterns.

. How can we use Logo to write a poem according to a selected pattern?

Aim: How can we write Logo procedures to print non-rhyming poems?

Development:

1. Write a sentence which will be used in the poem in two ways:

   . It will set the pattern for the other lines of the poem.

   . It will be included as the last line of the poem.

2. For example, if the sentence were:

   I am content.

   determine its parts of speech.

   I am content.

   (noun) (verb) (adjective)

3. Select lists of pleasing words which match the categories of your sentence.

   . Name each list appropriately according to the formula below.

   . Write a procedure which includes the names of the lists you have prepared. (Note: This lesson was written for Apple Terrapin Logo.)

```
TO WORD.SETS
  MAKE "NOUNS [DOVES CHILDREN KITTENS CLOUDS BLUEBIRDS FLOWERS BREEZES]
  MAKE "VERBS [LOVE SLEEP WELCOME SMILE SING ROLLICK PLAY]
  MAKE "ADJS [HAPPILY PEACEFULLY SOFTLY QUIETLY GENTLY WARMLY
              CHEERFULLY]
END
```

4. Write a procedure which chooses words at random from the lists above.

```
TO CHOOSE1
   OP ITEM (1 + RANDOM 7) :NOUNS
END

TO CHOOSE2
   OP ITEM (1 + RANDOM 7) :VERBS
END

TO CHOOSE3
  OP ITEM (1 + RANDOM 7) :ADJS
END
```

Applications:

1. Write a procedure which prints out the lines of a poem. Use the PRINT1 Logo primitive command. PRINT1 positions whatever is included within its brackets on the same line. We will use it to put a period at the end of each line.

```
TO VERSE
  PRINT1 (SE CHOOSE1 CHOOSE2 CHOOSE3 ) PR ".
END

TO BLANK.VERSE
  WORD.SETS
  REPEAT 3 [VERSE]
  PR [I AM CONTENT.]
END

TO START
  PR[]
  BLANK.VERSE
  PR[]
END
```

Summary:

1. How do we use Logo and pleasing language patterns to print out blank verse?

Assignments:

1. Edit the procedures above to make a poem in which the student supplies the first line of the poem instead of the last.

Assignments for Advanced Students:

1. Select another pattern which can print a title for each blank verse. Prepare procedures to print each blank verse with its own title.

TOPIC 30 - A

Logo Poems

Performance Objectives - Students will be able to:

. Edit Logo procedures which will print out rhyming verses according to selected language patterns.

Vocabulary - New: rhyme scheme
Review: language pattern,

Motivation:

. An important feature of many poems is the use of words in an arrangement in which words that sound alike (or rhyme) are used in interesting patterns. How can Logo help us to write these kinds of poems?

Aim: How do we write Logo procedures which will print out rhyming poems?

Development:

1. Select a rhyming scheme for your poem. (Note: This lesson was written for Apple Terrapin Logo.) For example, A-A-B-B or A-B-B-A.

2. Select a language pattern for each line of your poem. You might want to use a line of your own which could be included as the first or last line of each poem printed out by Logo. For example:

```
Line 1: No one will sleep  (A)
Line 2  (noun)  (verb)     (B)
Line 3  (noun)  (verb)     (B)
Line 4  (noun)  (verb)     (A)
Line 5  No one will sleep  (A)
```

Applications:

1. Edit the procedures in the last lesson to create lists of appropriate words for your pattern.

   . Some lists will be quite limited, since words will be to be selected not only for mood, and parts of speech, but also for rhyme. Here are a few suggestions for the verse shown above.

   Rhyming (A) words: peep, creep, leap, cheep
   Rhyming (B) words: ring, sing, fling, ping, zing

2. The following subprocedures may be used in this project:

```
TO WORD.SET
     MAKE "NOUNS [DOVES CHILDREN KITTENS BLUEBIRDS]
     MAKE "VERBSA [PEEP CREEP LEAP CHEEP]
     MAKE "VERBSB [RING SING ZING PING]
END


TO CHOOSEN
     OP ITEM (1 + RANDOM 4 ) :NOUNS
END

TO CHOOSEA
     OP ITEM (1 + RANDOM 4 ) :VERBSA
END

TO CHOOSEB
     OP ITEM (1 + RANDOM 4 ) :VERBSB
END

TO MY.LINE
     PR [NO ONE WILL SLEEP.]
END

TO VERSEA
     PRINT1 (SE CHOOSEN CHOOSEA ) PR ".
END

TO VERSEB
     PRINT1 (SE CHOOSEN CHOOSEB ) PR ".
END

TO RHYME.VERSE
     MYLINE
     WORD.SET
     REPEAT 2 [VERSEB
     VERSEA
     MYLINE
END

TO BEGIN
     PR []
     RHYME.VERSE
     PR []
END
```

. Call this program by typing BEGIN. Logo might print out a verse like this:

```
No one will sleep
Birds sing
Bells ring
Lambs leap
No one will sleep
```

Summary:

1. How have we used Logo to print out short rhyming verses according to selected language patterns and rhyming schemes?

Assignments:

1. Write procedures for other three-line, four-line, and five-line poems which include at least one line of your own in each poem.

Assignments for Advanced Students:

1. Write procedures for poems with a three-rhyme scheme, such as: A-B-C-C-A or A-B-A-C-C.

TOPIC 31 - M

Logo Conversations

Performance Objectives - Students will be able to:

- Edit a group of Logo subprocedures so they will be combined into a text/graphics program which simulates a conversation with the computer.
- Recognize a START subprocedure which calls all of the other subprocedures in a Logo program.

Vocabulary - New: format
Review: naming

Motivation:

The CHOOSE program can be edited, revised and expanded to include several more interesting choices during the simulated conversation with the computer. How can we edit the program to do this?

Aim: How can we include more choices of pictures and other options in the Logo conversational program (CHOOSE) we started in the previous lesson?

Development:

1. Prepare a plan or outline of the way you want the program to branch. (Note: This lesson was written for Apple Terrapin Logo.) This outline should be formatted so that the results of all offers, choices and options can be clearly read and understood. For example:

   . Which do you want to draw?

- a house	- a boat
If the answer is a house, Logo draws a house and roof.	If the answer is a boat, Logo draws a simple boat shape.
Do you want windows and a door?	Do you want a mast and a sail?
If the answer is 'Yes', Logo draws in windows and a door. Then the program stops.	If the answer is 'Yes', Logo draws a mast and sail. Then the program stops.
If the answer is 'No', Logo prints 'Bye' and the program stops.	If the answer is 'No', Logo prints 'Bye' and the program stops.
If any other choice is made, Logo prints 'Bye' and the program stops.	If any other choice is made, Logo prints 'Bye' and the program stops.

   - If any other choice is made, the program again offers the choice of a house or a boat picture.

2. In the outline formatted above, the student should observe that each branch terminates in one of three ways:

   . It stops the program.

   . It leads or returns to another subprocedure.

   . It returns to the beginning and re-starts the program.

3. The student should also observe that the entire program proceeds from one opening question and its set of choices.

   . The procedure which calls this beginning part of the program should be given a simple, standard title, such as START or SETUP, which can be easily spotted by anyone who wants to run the program.

<u>Applications</u>:

1. Each part of the formatted outline above will require at least one sub-procedure. (We can use the CHOOSE procedure in an edited form--see below). Note all primitives, abbreviations, and punctuation in the procedures below.

```
TO START
  CHOOSE
END


TO CHOOSE
  HOME CS HT PD
  PR [WHICH DO YOU WANT TO DRAW?]
  PR [PRESS H FOR A HOUSE]
  PR [PRESS B FOR A BOAT]
  MAKE "KEY1 RC
  IF :KEY1 = "H HOUSE STOP
  IF :KEY1 = "B BOAT STOP
  START
END


TO HOUSE
  BOX
  FD 50 RT 30
  ROOF
  HOME
  GO.ON
END
```

```
TO BOX
  REPEAT 4[FD 50 RT 90]
END


TO ROOF
  REPEAT 3[FD 50 RT 120]
END

TO BOAT
  LT 45 FD 50
  SETH 90 FD 100
  RT 135 FD 50
  SETH - 90 HOME
  GO.ON
END


TO GO.ON
  IF :KEY1 = "H PR [DO YOU WANT A DOOR?]
  IF :KEY1 = "B PR [DO YOU WANT A SAIL?]
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  MAKE "KEY2 RC
  IF :KEY2 = "Y CHOICES
  QUIT
END


TO CHOICES
  IF :KEY1 = "H CHOICE1
  IF :KEY1 = "B CHOICE2
END


TO CHOICE1
  IF :KEY2 = "Y DOOR.PANES
  PR[]
  FINISH
END


TO CHOICE2
  IF :KEY2 = "Y MAST.SAIL
  PR []
  FINISH
END
```

```
TO DOOR.PANES
  HOME
  RT 90 FD 15 LT 90
  DOOR
  HOME FD 40 RT 90
  PU FD 10 PD
  PANE
  PU FD 20 PD
  PANE
  PU HOME PD
END

TO DOOR
  REPEAT 2[FD 25 RT 90 FD 20 RT 90]
END
TO PANE
  REPEAT 4[ FD 10 RT 90]
END

TO MAST.SAIL              TO FINISH
  PU FD 35                  PR [THERE'S YOUR PICTURE.]
  PD FD 80                  PR [THANKS FOR THE CHAT.]
  RT 120                  END
  ROOF
END

TO QUIT
  PR [BYE!] STOP
END
```

<u>Summary</u>:

1. How should we proceed when we wish to include many choices throughout a program which simulates a conversation with the computer?

<u>Assignments</u>:

1. Edit the FINISH procedure so that it will print out and run a new choice:

    Do you want to start again?
    Press Y for YES
    Press N for NO

<u>Assignments for Advanced Students</u>:

1. Edit the CHOOSE procedure again to include a choice of either red or green pen color.

TOPIC 31 - A

Logo Conversations

Performance Objectives - Students will be able to:

- Identify elements in an interactive Logo program which permit the user to simulate a conversation with the computer.

Vocabulary - New: IF, THEN, ELSE.
Review: interactive, REQUEST (RQ).

Motivation:

- Computers are used for games as well as instruction and business. How can we write a game which will include a conversation with the computer?

Aim: How shall we combine Logo procedures into a simple word game?

Development:

1. Prepare an opening procedure in which the computer greets the player, asks for the player's name, and invites the player to participate in a word game.

2. Include the greeting program in the START program which sets the game in operation.

   - There are several ways to do this; here is one. (Note: This lesson was written for Apple Terrapin Logo.)

```
TO START
  GREET
  QUIT
END

TO GREET
  CLEARTEXT TEXTSCREEN
  PR [HELLO! PLEASE TYPE YOUR NAME.]
  MAKE "NAME RQ
  PRI]
  PRINT1 [HI, ] RQ
  PR []
  PR [I HOPE YOU WANT TO PLAY MY WORD GAME.]
  PR [HERE IT IS!]
  WORD.GAME
END
```

3. **Write a procedure for the game in which the player's selection is matched with the computer's selection. If the selections match, the player wins. If they do not, the computer wins.**

   . Again, there are several ways to do this. In the procedure below, the computer makes a random selection from a list.

   . Then the player's selection and the computer's selection are compared (letter by letter) to see whether or not they match. The equal sign (=) is used to make the comparison.

   . This test is included in the conditional IF/THEN/ELSE command. All are used in this program to instruct Logo to do the following:

   - Examine the two words;

   - IF they are the same, THEN print a congratulatory message,

   - ELSE (if they are different) print a consolation message.

```
TO WORD.GAME
  PR [ I AM THINKING OF A 4-LETTER WORD THAT BEGINS WITH A 'B' AND ENDS
     WITH A 'T'.]
  PR[]
  PR [ WHAT WORD DO YOU THINK IT IS?]
  PR[]
  PR [ TYPE IT -- THEN PRESS <RETURN>]
  MAKE "GUESS RQ
  PR []
  PRINT1 [ I WAS THINKING OF THE WORD: ] PR :OPT
  PR []
  IF :GUESS = :OPT THEN PR [GOOD WORK!] ELSE PR [SORRY!]
  PR []
  CHOICE
END
```

Applications:

1. The procedures above may be used with other subprocedures to complete the game.

```
TO WORD.SET
  MAKE "OPT ITEM [ 1 + RANDOM 14 ) [BAIT BEAT BEET BEST BOAT BOOT BOUT
       BUNT BUST BOLT BRAT BELT BENT]
  MAKE "OPT (SE :OPT )
END
```

```
TO CHOICE
  PR [DO YOU WANT TO PLAY AGAIN?]
  PR []
  PR [PRESS Y FOR YES]
  PR [PRESS N FOR NO]
  PR []
  MAKE "KEY RC
  IF :KEY = "Y WORD.GAME
END

TO QUIT
  PR [THANKS FOR PLAYING THE GAME!]
  PR [HOPE YOU ENJOYED IT!]
  PR [BYE!]
END
```

Summary:

1. How do we use Logo procedures to write a guessing game which includes a simulated conversation with the computer?

Assignments:

1. Edit the WORD.GAME subprocedure to include a counting subprocedure which gives the player three guesses before Logo calls the CHOICE subprocedure.

Assignments for Advanced Students:

1. Add the following lists to the WORD.SET subprocedure. Then edit WORD.SET so that Logo makes a random choice among the three lists.

   . Note: the 'hint' or 'clue' line in the WORD.GAME subprocedure will also need to be edited.

```
OP ITEM (1 + RANDOM 14) [FACT FAST FEAT FEET FELT FIAT FIST FONT FOOT
     FORT FLAT FLIT FRET FRAT]

OP ITEM (1 + RANDOM 12) [FAIL FALL FEEL FILL FOAL FOIL FOOL FOUL FOWL
     FUEL FULL FURL]
```

TOPIC 32 - E

The Wait Command

Performance Objectives - Students will be able to:

- Explain the effect of the WAIT command.
- Demonstrate an approximation of 5 seconds.
- Follow verbal directions to move FD and BK.

Materials: red squares, green squares

Vocabulary: approximate, FORWARD, BACK, REPEAT

Motivation:

- How do we determine a second?
- Can you approximate a second?

Aim: Approximately how long is a 5-second interval?

Development:

1. Play a game of counting off of five seconds (approximately). Put hands together, follow the teacher and count orally 1 and, 2 and, 3 and, 4 and, 5 and.

2. Play a "cha-cha-cha" game. Step: FD, BK, FD. Wait for five seconds, counting off the seconds orally. Then, step: BK, FD, BK.

3. Change the rules. "Cha-cha-cha" without counting off the WAIT time. Repeat the steps when you believe the five seconds have passed. Practice in pairs. Work up to a chorus line.

4. Play a color change game. Hand out a set of cards to each student. The order of the cards from top to bottom is red, green. Switch colors at 5-second intervals. Try switching after an oral count. Then, switch without using the count.

5. Have students go to the computer. Use a primitive command to color the background, or use the C key (see Keystrokes) and the variable.

6. Introduce the WAIT command as a seconds counter. Add the WAIT command to the Keystrokes, if it is necessary. Have students experiment with numbers from 100 to 300 to see which WAIT interval best suits their purposes.

Applications:

1. Students will determine the number of 5-second intervals they can count off as the second hand on the clock moves from the 12 back to the 12.

2. Draw pictures of a shape as it goes through changes of direction, color, and shape. Write out the commands for these changes.

   . Use the REPEAT command as a way of reducing the task.

3. Create games in which there is a wait interval. For example, a student might hold a ball until an exact X-second-interval has elapsed before he/she passes the ball along to the next child on the line.

Summary:

1. Can you demonstrate periods of time in seconds?

Assignments:

1. Have the students write simple programs that use the WAIT command.

TOPIC 32 - M

The Wait Command

Performance Objectives - Students will be able to:

- Explain the function and use of the WAIT.
- Understand the time measurement of WAIT.
- Use the WAIT command to create simple animations.

Material: blank chart, felt marker

Vocabulary: second, delay, approximately

Motivation:

- Understanding the WAIT command, which delays the sprite's movement between two other commands.
- Chart the student's responses to the question of how fast the sprite on command moves from zero speed.

Aim: How does an understanding of the time span of WAIT improve animation?

Development:

1. Determine the span of one minute by counting 1 and 2 and 3, up to 60.
2. Have students look at the second hand on the clock as it moves through one minute.
   - Have the students try counting orally with the clock through one minute.
3. At a given direction, heads will go down on desks. Each student will raise his/her head when one minute has passed. Accuracy will be checked against the clock. Stay with the test through at least three tries.
4. Explain that each of these counts of "1 and" is approximately 1 second.
5. Explain that the computer waits in parts of a second, or that the computer has counted up through 60 by the time they have counted 1 and.
   - What will the computer's count be when you have counted through 60?
6. Discuss whether the delay in movement on a command of WAIT 1 could easily be seen. What might you type in so that you could easily see the delay in movement?

7. Then have a student type in the following:

```
TELL 1
SETCOLOR 1
CARRY 1
SETSPEED 0
HOME
```

8. Have the students type in one line of commands such as the following:

```
FD 40  WAIT # BK 40.
```

9. Try to arrive at numbers where the effect is negated by the speed of the delay. Have the students decide if they would want to use these numbers.

Summary:

1. Judging from the time interval of the WAIT command, can we come to some general idea of the FD/BK speed of a sprite from a speed 0?

2. Which of your responses to how fast the sprite moves from speed 0 comes closest to what we have determined? (Check through the chart to determine the best responses.)

Assignments:

1. Research what things happen in our world or in our universe at incredible speeds, or in split-second time.

2. Using the WAIT command, write programs that cause delays in changing the color of the background, the shape of the sprite, and so on. Illustrate the changes.

3. Research and write about any animal that changes color or shape for specific purposes.

TOPIC 33 - M

Making New Shapes

Performance Objectives - Students will be able to:

- Explain how to create their own shapes.
- Make new shapes with sprites.
- Differentiate the keys that move the cursor in MAKESHAPE.
- Move their own shapes around the screen.

Vocabulary: SPRITE, MAKESHAPE

Motivation:

- Would you like to see the truck and the plane go in other directions than they go when you enter LOGO? Would you like to have more shapes to work with?

Aim: How is it possible to make other shapes?

Development:

1. Type into the computer MAKESHAPE 1. What do you see? (A 16 by 16 grid of the plane.) Then, try MAKESHAPE 2, MAKESHAPE 3, etc.

2. Type into the computer MAKESHAPE #. What do you see? There are some shapes already in the computer's memory, but the MAKESHAPE # you have now called up shows a blank. You can use this blank grid to make a shape of your own.

3. Demonstrate how to use the cursor to blacken the squares in the grid. Let the students work with the keys that move the cursor right, left, up, and down.

4. You are now ready to have a Sprite carry your shape. Example:

```
TELL SPRITE 1
CARRY #
SETCOLOR 1
SETHEADING 90
HOME
SETSPEED 30
```

   . Note: The # gives the sprite the original shape you have just created.

5. Now change the variables of the sprite.

6. Give each student a 16 by 16 grid. Have them color in the boxes needed to create the designs of their choice.

7. Have the students create these designs with the MAKESHAPE command. (Use any number above the last number used by LOGO.)

8. You are now ready to have these shapes carried by sprites. (Each sprite can carry only one shape at a time.)

9. Experiment using all the sprite commands learned.

   . Record your programs and grid designs.

Applications:

1. Students will use Turtle Mode and Sprite Mode to create scenes (landscapes, seascapes, and others).

   . Example: Use turtle lines to make a road that a truck can drive down.

Summary:

1. How does the ability to increase the number of shapes enhance the writing of programs?

Assignments:

1. Divide off 16 x 16 areas on graph paper and draw shapes.

2. Write some programs which incorporate some of these shapes.

TOPIC 34 - E

Magnifying a Shape

Performance Objectives - Students will be able to:

- . Demonstrate a readiness for creating enlarged shapes.
- . Create an enlarged shape on grid paper.

Material: 16 by 16 grid sheets

Vocabulary: MAKESHAPE, brackets, enlarge

Background:

- . Review the commands FORWARD, BACK, LEFT, RIGHT.

- . Review talking jointly to sprites.

Motivation:

- . Would you like to have larger images appear on the screen?

Aim: How do we get bigger shapes on the monitor?

Development:

1. To create a box that is twice as tall and twice as wide as the box usually seen on the monitor, type in the following:

```
TELL [1 2 3 4]
CARRY :BOX
SETCOLOR #
HOME


TELL 1
FORWARD 16


TELL 2
FORWARD 16
RIGHT 90
FORWARD 16


TELL 3
RIGHT 90
FORWARD 16
```

2. We can make the sprites which make up the box move together as long as we speak the command to all the sprites at the same time.

```
TELL  [1 2 3 4]
SETHEADING 90
SETSPEED 50
```

3. Can we get a larger rocket the same way? Let's see.

```
TELL [1 2 3 4]
CARRY :ROCKET
SETCOLOR #
HOME

TELL 1
FORWARD 16

TELL 2
FORWARD 16
RIGHT  90
FORWARD  16

TELL 3
RIGHT 90
FORWARD 16
```

4. Explain that the rocket doesn't look solid because unlike the box, it doesn't have every space blackened in the Makeshape grid.

5. Use the command Makeshape and produce first the box in its grid and then the rocket so that the students can see the difference.

6. Explain that they will have to enlarge the rocket from their own designs which they will make in Makeshape. But first they should draw their own shapes on paper.

7. Pass out two grids to each student. Tell them they are to make the rocket longer by drawing their shapes in the grids. Keep the rocket in Makeshape as the students work.

<u>Applications</u>:

1. Have students key in their shapes. Then, have them command the sprites to carry their shapes.

<u>Summary</u>:

1. Why is it necessary to use more than one grid if we want a larger image on the monitor?

2. How do we get the enlarged figure to stay together when it moves?

Assignments:

1. Paste together four 16 x 16 grids and then draw the largest ball possible inside the grid so that it touches the grid in four places.
2. Separate the smaller grids, color the parts of the ball in each quadrant and insert the grids in the Makeshape booklets for future use at the computer.

TOPIC 34 - M

Magnifying a Shape

Performance Objectives - Students will be able to:

- Demonstrate the way an image can be enlarged beyond the 16 by 16 grid.

Background:

- Review the skill in keying grid images in Makeshape.
- Review the coordinate commands.

Vocabulary: coordinates

Motivation:

- Are you always satisfied with the size of the image the sprite carries on the screen?

Aim: How do you MAKESHAPE to enlarge the images that appear on the screen?

Development:

1. Type:

```
TELL 1
CARRY :BALL
SETCOLOR #
HOME
```

2. How can you make the shape bigger?

3. Then, type in Makeshape 5, leaving it on the monitor as you pass out to each child a 16 by 16 grid. Have them shade in the ball on their grid. Then repeat the question: How can you make the shape bigger?

4. If there is no correct answer, explain that a second grid can be used to enlarge the shape. Type in Makeshape 6 and shade in one half of a sphere, using as much of the grid as you can. Then type in Makeshape 7 and shade in the bottom half of a sphere, using as much of the grid as you can.

5. Type:

```
TELL 2
CARRY 6
SETCOLOR #
HOME
```

6. Type:

```
TELL 3
CARRY 7
SETCOLOR #
HOME
```

7. Set Makeshape 6 atop Makeshape 7 by using coordinate commands, talking to each sprite separately as you line up the half spheres. (Note: the ball appears only slightly larger, but the method is learned.)

Activities:

1. Pass out four 16 by 16 grids and tell the students to paste them together and draw a shape (other than a ball or square) that will use the four grids.

2. Students should have an opportunity to draw the completed shapes on the computer.

3. Have them figure a way to make an enlarged shape made from several makeshapes to move as one shape.

Summary:

1. How does the use of coordinates make it easier to fit together the parts that make up an enlarged image?

Assignments:

1. Construct Makeshape booklets. The booklets should include blocks of 16 by 16 grids. Students should be encouraged to create shapes for scenes.

TOPIC 35-E

Total Turtle Trip Setheading

Performance Objectives - Students will be able to:

- Follow verbal directions to move around simple closed figures.
- Describe a simple closed figure.
- Define the concept HEADING.
- Define a total turtle trip.

Vocabulary - Review: FORWARD (FD), RIGHT (RT) and LEFT (LT) commands

Materials: String, tape, set of cards with one of the following headings (0, 90, 180, 270, 360) written on each card along with the command SETHEADING, geoboards.

Aim: Along what kind of path does the turtle move if it is to make a total turtle trip?

Motivation:

- Let's consider some trips we ourselves take: around the block; to the refrigerator for a glass of milk and back to the table; when we line up for the sliding board, slide, and come back to line up again; when we successfully make it around all the bases and back home in playing baseball, and so on.

- What is common to all these trips? (beginning and ending heading the same.)

- In making all these trips, we start out from and return to the same spot. Turtle often starts out from a given location (a heading) and returns to the same heading.

- Today, we are going to look at some of the turtle's trips to decide when it has made what we will learn is a total turtle trip, and when it has not. Then, we can look at some of our own trips and decide which of them are total turtle trips.

Development:

1. Have the students help in taping to the floor simple closed figures made with the string.

2. Discuss the lines made with the string: They cross over one another; they do not.

3. Explain: If the turtle draws a figure without crossing over any of its lines, and returns to its original heading, then we say that the turtle has closed its path, or has made one total turtle trip.

4. Choose a pair of students. One will walk closely around the edges of a floor figure as the other gives the correct commands to change the state of the turtle, giving the turns as right turns.

   . The child who is walking the trip will assume the startup heading of the turtle on the monitor. The commands should be given relative to objects in the room or in the LOGO language.

   . Give each of the remaining students a set of five cards, each of which has a heading and the command SETHEADING written on it. As the student giving the commands calls out the commands for the right turns, the other students should determine the correct HEADING card to hold up.

   . Note: The teacher should hold up the correct card if necessary.

   . Were any lines crossed over?

5. Have the students note the beginning heading and the end heading of "our" turtle.

   . Make it clear that the turtle can begin its trip from any heading. However, it must end where it began if it is to complete a total turtle trip.

6. Explain: As the turtle rotates, it passes through 360 numbered locations on an invisible circle. They will learn them all as they get to know the turtle better, but as of now they know HEADING 0, HEADING 90, and HEADING 360. The positions will be easy to learn as they are numbered in order, beginning 0, 1, 2,...around the circle.

7. Count how many times the turtle (child) "closed" its path.

8. Explain: If the turtle closes its path once, it has turned once totally -- rotated clockwise or counterclockwise back to its original heading.

9. Demonstrate the circle through which the turtle has passed by having a student walk around the floor figure, with the only contact with the figure coming at the vertices. Tracing that figure with the string should reveal the circle.

Applications:

1. Have the students walk through other simple closed figures and ask the same questions, discuss the same concepts as those above.

Summary:

1. Describe the path the turtle makes when it walks a total turtle trip.

2. Name as many simple closed figures as you know.

3. In which direction is the turtle facing when it has turned once totally?

Assignments:

1. Construct on geoboards as many of the closed figures named in the summary as possible. Then, copy on paper those you believe can be programmed using only the headings given in this lesson.

2. Write out the programs for the figures chosen in Assignment 1.

TOPIC 35 - M

Total Turtle Trip Setheading

Performance Objectives - Students will be able to:

- Write a program for a simple closed figure.
- Rotate a simple closed figure through total trips.
- Define a total turtle trip.
- Contrast the use by the turtle of the commands SETHEADING and RIGHT in rotating through its path.

Materials: cardboard squares with one-half of the turtle drawn in the lower left-hand corner, paper, rulers, protractors, crayons

Vocabulary - Review: HEADING, SETHEADING

Motivation:

- With which of the following commands, can we draw a simple closed figure?
  - RIGHT 90 FORWARD #
  - SETHEADING 90 FORWARD #
  - SETHEADING # FORWARD #

Aim: How do the commands RIGHT and SETHEADING differ in effect?

Development:

1. Send two students to the computer. Have one verbalize the procedure Square while the other types the procedure. It should read like this:

```
TO SQUARE
  FD 40
  RT 90
  FD 40
  RT 90
  FD 40
  RT 90
  FD 40
  RT 90
END
```

   . Did the turtle rotate back to its original heading?

2. Next, hand out a cardboard square, paper, and crayons to each student. Have them rotate the square from its original heading 0 through 90 degrees four times. Each time have them trace around the square onto the paper.

3. Now, you have made a different design, but after making the design, did the turtle return to its original heading? How do you know this?

   - Explain: The turtle rotated through its 360 headings, closed its path, returned to its original heading, and so made a total trip.

   - Let the students color in the four squares that make up the design.

4. Have students verbalize in two ways the procedure they have just simulated:

   - SETHEADING 90 SQUARE
     SETHEADING 180 SQUARE
     SETHEADING 270 SQUARE
     SETHEADING 360 SQUARE

   - REPEAT 4 [RT 90 SQUARE]

5. Have students explain the difference between RT 90 and SETHEADING 90.

   - HEADING 90 is one location only on the turtle path, to which the turtle rotates any time it is given the command SETHEADING 90.

   - The 90 of RIGHT 90 is how many degrees the turtle adds to the heading it is at when the command RIGHT 90 is given.

   - So, if the turtle is at heading 30 when it is given the command RIGHT 90, to which heading would it rotate?

   - If the turtle at heading 30 when given the command SETHEADING 90, to which heading would it rotate?

6. Have the students place the cardboard square on a diagonal on the top right square of the design they traced on their paper. Determine the heading.

   - When it is determined that the heading is 45, have the students write a command which rotates the square from a heading of 45.

     - SETHEADING 45 SQUARE

     - RT 45 SQUARE

7. Work together to write a procedure that rotates the square through successive 45-degree angles toward a total turtle trip. Decide which of the commands above can be used.

Applications:

1. Write the procedures for a triangle, rectangle, and a pentagon. Then, write a procedure for each of these closed figures to rotate the figures through a total turtle trip.

Summary:

1. Why is it important to understand the difference between the commands Right (any given number) and SETHEADING (any given number).

2. Why must one exercise extreme care when using the setheading command?

Assignments:

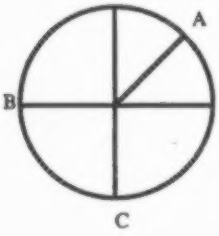
1. Write programs for a hexagon and an octagon and test the programs on the computer.

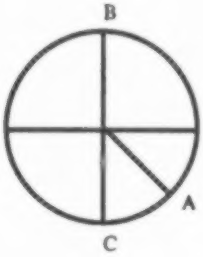
2. Work through the attached worksheets and make your own to pass around.

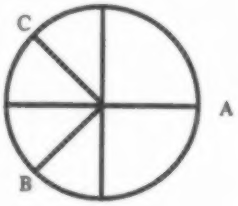
WORKSHEET 1

TOPIC 82M
TOTAL TURTLE TRIP SETHEADING

Which is the "heading" for each labelled A, B, and C location? Write your answers below.







A ________	A ________	A ________
B ________	B ________	B ________
C ________	C ________	C ________

1. If the turtle is at heading 60 when it is given the command RIGHT 90, to which heading would it rotate? ______________

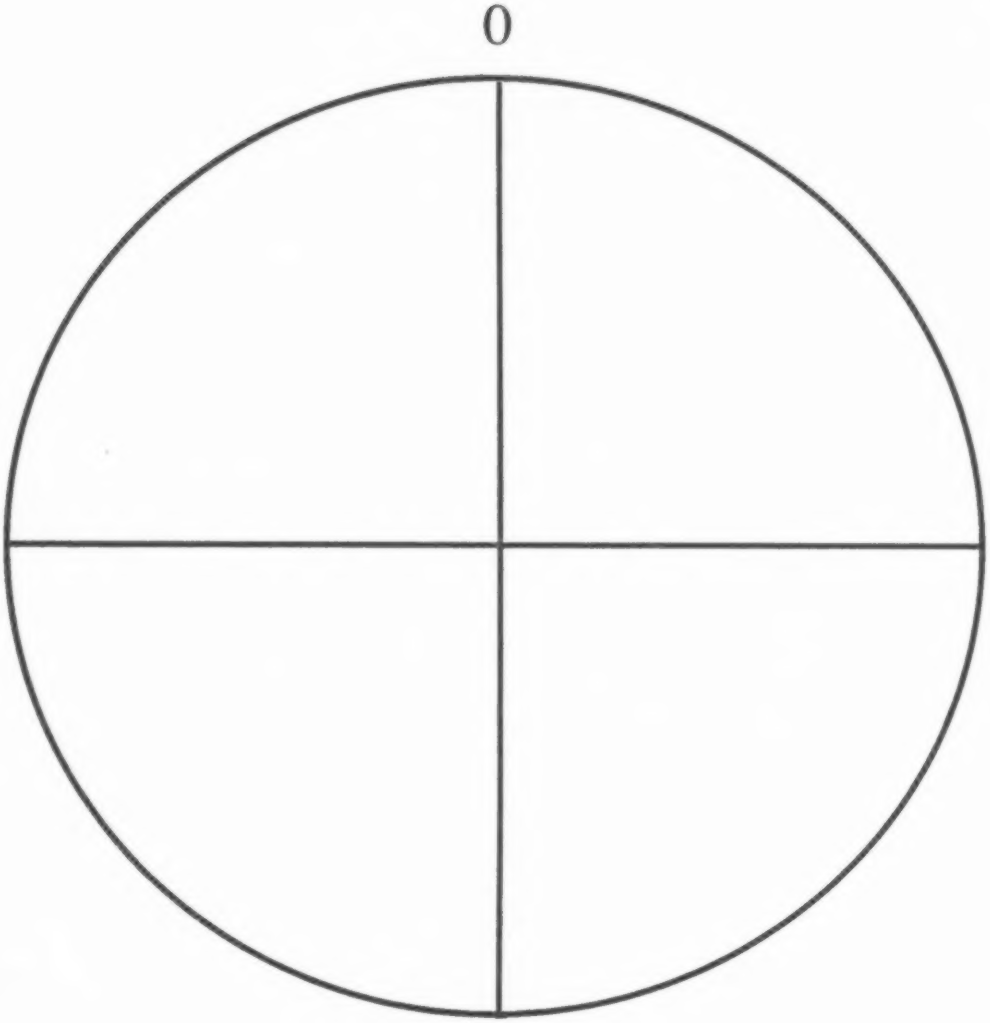
2. If the turtle is at heading 60 when it is given the command SETHEADING 90, to which heading would it rotate? ______________

3. If the turtle is at heading 60 when it is given the command LEFT 90, to which heading would it rotate? ______________

4. If the turtle is at heading 60 when it is given the command SETHEADING 30, to which heading would it rotate? ______________

WORKSHEET 2

Mark off the circle in "turns" of 10 and label the locations to 360. You may use a protractor.



TOPIC 36-E

Setheading with a Variable

Performance Objectives - Students will be able to:

- Define a variable.
- Name some LOGO commands that take variable.
- Create designs, using SETHEADING/variable.

Vocabulary: variable

Materials: chalkboard, paper, pencils

Aim: Why is the variable useful?

Background: Review the format for a procedure.

Motivation:

- Do you know how the command SETHEADING allows you to type in any number you want to, up through 360? (responses)

- Well, the programmer who wrote the SETHEADING procedure left a blank spot in its memory so that the command can't work unless you, the user, fill in the blank with a number that fills in the blank in SETHEADING's memory.

- We call that number a "variable." Usually, it's lots of fun to play around with variables.

Development:

1. Let's make a crazy design. If we play around with SETHEADING with a blank spot in its memory, we might make some really crazy designs.

   - First, we have to give our blank spot in SETHEADING's memory a name. The name can be a letter or a word. Let's keep its name simple by using only the letter "X".

   - We also want movement, since all SETHEADING does is set the turtle so it is ready to head off. Let's use FD 20.

   - Next, we will have to give our design a name -- maybe "Mad," since our design will be crazy, and "mad" is easier to type than "crazy." We are going to write a procedure Mad, which will give LOGO a brand-new command. (The teacher will write the procedure on the blackboard.)

2. Mad has to know right off that SETHEADING has a blank spot in its memory, which we can fill in with a number (variable) later. "X" is the name we have given to the blank spot. So, we say TO MAD with an "X". But since "with an X" is too much to type, we would type only :X and say "dots X."

```
TO MAD :X
  SETHEADING :X
  FD 20
END
```

3. Use the new procedure as a LOGO command. Fill in the procedure's blank spot (the "X") with a number. Type:

```
MAD 10
MAD 25
MAD 100
```

4. If we typed in SETHEADING 10 FORWARD 20, would we have gotten the same effect as MAD 10?

   . Let's check. We will clear the screen and type in:

```
MAD 10

SETHEADING 10 FORWARD 20
```

5. What does this tell us about SETHEADING?

   . If the responses are off target, try the following:

   - Type SETHEADING and press ENTER. Next, MAD and press ENTER.

   - Why is SETHEADING giving us an error message? (Mad requires a variable to fill in that blank spot in SETHEADING's memory.

   - Why is SETHEADING giving us the same error message? (It has a blank spot in memory in its procedure.)

6. Look at the procedure Mad on the board. We know that Mad and SETHEADING have blank spots in memory in their procedures. Could we say that Forward has a blank spot in memory? Can we check to find out?

```
Type FD   ENTER
```

   . Did we get the same error message as above?

7. Have the students supply a letter name to the blank spot in Forward's memory in its procedure. Use a letter other than X, unless you would like to have the same variables for each command.

8. Does Mad have to know that there will be two blank spots, one for each of the commands, in its procedure?

```
TO MAD1 :X :Y
  SETHEADING :X
  FORWARD :Y
END
```

Applications:

1. Have students copy the procedure Mad in their LOGO books for practice when they come to the computer.

   . Students can copy their designs onto drawing paper for the bulletin board.

Summary:

1. What is a variable?

2. Why are we using the command MAD when SETHEADING and FORWARD together will do the same job for us?

Assignments:

1. Write a procedure that uses the MAD command at least five times.

2. Write a procedure that uses the MAD1 command at least five times.

3. Copy the results of the two above assignments on paper, overlaying the designs.

TOPIC 36 - M

Setheading with a Variable

Performance Objectives - Students will be able to:

- . Follow verbal commands to SETHEADING.
- . Use a variable to change a heading.
- . Define a subprocedure.
- . Define an input.

Vocabulary: variable, input, subprocedure

Background:

- . Review the commands PENUP, PENDOWN
- . Review the headings of the invisible circle the turtle travels through

Motivation:

- . What does it mean when you are playing "Hide-and-Seek" and someone says, "I'm heading for that tree?" (Responses)

- . The turtle in setting its heading gets itself ready to head (go forward) to a specific point on its invisible circle.

- . Let's see if turtle can play "Hide-and-Seek."

Aim: Can you change a heading without having to write a new procedure?

Development:

1. Select seven volunteers.

   - . Assign one to be "It." Upon the command Forward, the others take preassigned floor positions on a straight line away from It.

   - . It then heads out to them. Once contact with any one of the six is made, it returns to base before heading out again.

2. The same process can be used with the turtle.

   - . To do this, we need six different headings for the six students who hid.

   - . We could write the same steps six times, or we could write one set of directions, using a variable.

3. **Explain: When you write a procedure and you put in (or input) a variable, you are filling a blank spot in the memory of LOGO.**

   - **You can name the blank spot with a letter or a word.**
   - **If you want to use this procedure in a turtle drawing, you will have to write something in that blank spot, or LOGO is unable to do the procedure.**
   - One advantage to using a variable is that you can use just one procedure, but type in many different inputs.

3. Help the students verbalize the Hide-and-Seek game.

   - Call the procedure "Hide".
   - Since there are six different headings, use the variable with the heading.
   - Label the variable (the blank spot in memory in the procedure) "X."
   - Tell the Hide procedure right at the beginning that it needs the variable "X."

4. Develop a facsimile of the procedure on the chalkboard. Begin with:

```
TO HIDE with an X
  SETHEADING with an X
  FORWARD #
```

   - As the procedure develops, substitute "with an X" with ":X".
   - Ask the students how the turtle can go forward without leaving a trail.
     - Elicit responses of PU and PD.
   - Edit the procedure accordingly.

```
TO HIDE :X
   SETHEADING :X
   PU
   FD #
```

   - Edit the procedure to leave a little speck at this heading to see where the ones are hiding by adding the commands PD and FD #.
     - Note: If you use MAKESHAPE and sprites, you have no problem with leaving a trail or speck.
   - Edit the procedure so that the turtle returns to its base (HOME). Remember not to leave a trail.

5. Have a student type in the procedure as another reads it from the board.

```
TO HIDE :X
  SETHEADING :X
  PU
  FD 40 (Agree upon a number)
  PD
  FD 2 (Use some low number.)
  PU
  HOME
  PD
END
```

. Now HIDE can be used with any heading simply by putting any number in the blank memory. Test the procedure with the following:

```
HIDE 15                        HIDE 90
HIDE 135                       HIDE 225
HIDE 270                       HIDE 315
```

6. Write a procedure for "It" called SEEK.

. Since SEEK has the same six headings, it also needs the blank spot in memory designated as X.

. ˜It's" trail in SEEK should be visible so the students know it actually found the six players who hid.

. It must also return home each time before it heads out again. The procedure should look like this:

```
TO SEEK :X
  SETHEADING :X
  FD 42
  HOME
END
```

7. Put the two procedures together so they will work one after the other.

. Create a procedure that puts both HIDE and SEEK to work. Call it HIDE.AND.SEEK.

- Both procedures (HIDE and SEEK) are now subprocedures.
- A subprocedure is a procedure that is "put to work" by another procedure.

- If we use the variables from the HIDE practice, which must also be the variables for SEEK, then HIDE.AND.SEEK will have no blank spots in its memory.

```
TO HIDE.AND.SEEK
  HIDE 15
  HIDE 90
  HIDE 135
  HIDE 225
  HIDE 270
  HIDE 315
  WAIT 300
  SEEK 15
  SEEK 90
  SEEK 135
  SEEK 225
  SEEK 270
  SEEK 315
END
```

Applications:

1. Have students practice using the procedures HIDE and SEEK, since they can easily change the headings here, but not in HIDE.AND.SEEK.

2. Since HEADING takes a number input and numbers can be added or subtracted, the computer can add and subtract.

   - Simply tell the computer to do :X + # for each number change.

   - Call this easier procedure HIDE1 :X.

     - The computer will then add on the # to the original number, and then the # to the last given number.

   - Why will this new procedure "drive the computer mad?"

   - Do you know how to rescue the computer from its "madness"? Tell the students to get together and work on these questions.

Summary:

1. What is a variable?

2. If a procedure has variables, where should you first let this be known?

Assignments:

1. Name the LOGO commands other than SETHEADING that use variables.

2. Write a procedure that uses variables with some of these commands.

   . Use as many of these commands as possible in one procedure.

   . Note: Use small numbers to insure that you stay on the screen.

TOPIC 37 - E

Setheading with Conditionals

Performance Objectives - Students will be able to:

- Use the IF/THEN conditional successfully in oral programming.
- Demonstrate an advanced control of turtle.

Vocabulary - Review: SETHEADING, FORWARD, BACK

Motivation:

- Wouldn't it be nice if the turtle could check itself every so often to see if it is following directions?

Aim: How can we program the turtle to check for accuracy as it proceeds?

Development:

1. Play "Spin the Turtle."

   - Clear an area two feet from a chosen center (Home). Decide on positions for SETHEADING 0, SETHEADING 90, SETHEADING 180, and SETHEADING 360. Place masking tape in a direct line from each position to two feet from the center.

   - Have a student spin a few times through the different "headings" while standing in the Home position. Then, blindfold the student.

   - Have the other students take turns programming the turtle to spin. After giving any of the above SETHEADING commands, ask the "turtle" to test if he has reached that heading.

     - If he has, he is to stop.

     - If he has not, he is to continue spinning; but he is not to go beyond the heading given as the test.

     - If the programmer fails to set the conditions for the test, the "turtle" can "balk," and the programmer must "debug" or lose a turn.

2. Example: The programmer says:

```
TURTLE
SPIN
SETHEADING 90
test IF your heading has reached 90.
IF it has, STOP.
IF it has not, THEN SPIN until it does.
```

- The student at center is started each time at SETHEADING 0. The student at center is "turtle" only as long as he or she can spin and stop correctly at the position given.

- Students should be rewarded for "mission accomplished."

3. The following variations are of greater difficulty.

   - Programmers may decide to have "turtle" spin to halfway headings before reaching the heading beyond which it may not go.

     - Example: The programmer says:

```
TURTLE
SETHEADING 45
test IF your heading has reached 90.
IF it has, STOP.
IF it has not, SPIN from the heading you have now + 45.
```

   - Programmers may decide to have "turtle" move out two feet, then back two feet as it moves through halfway headings.

     - Example: The programmer says:

```
TURTLE
SETHEADING 45
FORWARD 2
BACK 2
test IF your heading has reached 90.
IF it has, STOP.
IF it has not, SPIN from the heading you have now + 45.
```

4. Begin substituting the commands to those that the turtle understands.

   - Explain that "test" is not necessary because the turtle automatically tests whenever it sees the conditional IF.

   - Explain that "your heading" is ":HEADING" on the computer.

   - Explain that the phrase "has reached" becomes "=" (the equal sign).

5. Once substitutions are made, the program above becomes:

```
TURTLE
SETHEADING 45
FORWARD 2
BACK 2
IF :HEADING = 90, STOP
SPIN :HEADING + 45
```

Summary:

1. How does the IF/THEN test give you greater control in programming?

Assignments:

1. Add other floor positions that the students can successfully spin through.

TOPIC 37 - A

Setheading with Conditionals

Performance Objectives - Students will be able to:

- Use the IF/THEN conditional successfully.
- Demonstrate an advanced control of the turtle.

Vocabulary: conditional, variable, plus (+)

Materials: Rexo sheets of circles with the center, and vertical and horizontal diameters defined; rulers

Motivation:

- Your mother sends you to the store to buy a product only if its cost is within a given cost. How do you go about checking if you should buy it?

- You want to instruct the turtle to move only within a given range and have him check the range as he moves. How can you do it?

Aim: How can you control the turtle as it moves through its sequence of headings?

Development:

1. Hand out the circles and have the students follow these directions at the computer to fill in the radii:

```
FORWARD 30
BACK 30
SETHEADING 10
```

2. Repeat the above commands until it reaches SETHEADING 360.

   - Increase the heading each time by 10.

   - Draw the resulting radii in corresponding positions on the rexo sheets.

3. When the students have completely filled in their rexo sheets, ask them:

   - What were you asked to do in this exercise?

   - How could you have shortened the task and still have kept the turtle within the limits?

   - If someone suggests the command REPEAT, explain that while it works, another extremely useful control is the IF/THEN conditional.

. Then, type the following:

```
TO SPOKE :HEADING
  FORWARD 30
  BACK 30
  SETHEADING :HEADING
  IF :HEADING = 360 STOP
  SPOKE :HEADING + 10
END
```

Summary:

1. How does the IF/THEN conditional give you greater control in programming?

Assignments:

1. Allow the students time to work in pairs to work out programs of their own that use the IF/THEN conditional.

TOPIC 38 - E

Logo Sounds and Music

Performance Objectives - Students will be able to:

- Use LOGO to create simple rhythms.
- Demonstrate various tempos using short and long sounds.

Aim: Can we use Logo to make sounds and music?

Teacher Background:

- The following set of three LOGO Sounds and Music lessons are designed to be used with the Commodore 64 version of Terrapin LOGO. In order to access the music capability of C-64 LOGO, one must "READ" the "MUSIC" file from the C-64 Logo Utilities disk. Proceed as follows:

  - Boot the C-64 LOGO disk.

  - Remove the C-64 LOGO disk and insert the C-64 LOGO Utilities disk.

  - Type: `READ "MUSIC` (press RETURN)

Motivation:

- Play a musical version of "Follow The Leader" with the class.

- Have a student, the leader, clap a short repeating sound with both hands. Tell the class that they are to listen very closely to the sounds they will hear. When the student stops clapping, the class will try to imitate the exact sounds they heard.

- Choose various students to be the leader. Any pattern of sounds is acceptable. As the students become better at imitating the sounds they hear, the rhythms should become more complex.

Development:

1. Explain that a sound (or beat) that continually repeats itself produces a rhythm. If you began your lesson with Follow The Leader, you may tell the class that they were creating rhythms with their clapping.

2. The time between the beginning of one sound and the beginning of the next sound is called the duration of that sound. The LOGO sound has been pre-programmed to imitate that of a snare drum.

3. Make a short banging sound.

```
SSH  5     (press RETURN)
```

. Have the students orally imitate the sound they just heard.

. Type the following: `SSH  10` (press RETURN)

. Why was the sound longer? Elicit that the greater the number, the longer the duration of the sound.

4. Let students experiment with multiples of 5:

```
SSH  5      (press RETURN)
SSH  10     (press RETURN)
SSH  15     (press Return)
SSH  20     (press RETURN)
SSH  25     (press RETURN)
```

5. Explain that it is possible to repeat a sound more than one time, thereby producing a rhythm.

. Examples:

```
short    short     short     short     (sound)

SSH 5    SSH 5     SSH 5     SSH 5    (press RETURN)


short    long      short     long      (sound)

SSH 5    SSH 10    SSH 5     SSH 20   (press RETURN)
```

6. Make up different combinations of rhythms using short and long durations.

7. Let the students play the game "Morse Code Music." Directions:

. Using SSH 5 (.) and SSH 20 (_), allow the students to type in the following rhythms:

a) . . _ _ . .

b) _ _ . _ _

c) _ . _ . _ .

d) . . . . _ _ _ _

Summary:

1. Which LOGO procedure can be used to create sounds?

2. How do numbers control a sound's duration?

Assignments:

1. Type out the rhythmic pattern (using SSH 5 and SSH 20) to "Mary Had A Little Lamb."

TOPIC 38 - M

Logo Sounds and Music

Performance Objectives - Students will be able to:

- . Use the REPEAT procedure to condense groups of long and short sounds.
- . Identify time values (musical notation) through rhythm patterns.

Aim: Can we use Logo to produce rhythms?

Motivation:

- . Review fractional parts of a circle. Look around the room for objects that are circular.
- . Give each student three paper circles. Ask them to cut one circle into two equal parts, one circle into four equal parts and one circle into eight equal parts.
- . Tell them to imagine that each circle represents a whole pizza pie. Ask, "Which pizza has the greatest number of pieces?" "Which pizza has the least number of pieces?" Ask, "Which piece would you rather eat?"
- . Elicit from the students that the greater the number of equal pieces from one pizza (whole), the smaller each piece becomes.

Development:

1. Review work from previous lesson:

   . making short sounds (SSH 5)

   . making long sounds (SSH 20)

2. Tell the students that there is an easier way to produce long rhythms without typing SSH over and over again.

   . Introduce (or review) the REPEAT procedure.

3. Demonstrate the relative ease with which the REPEAT procedure produces rhythms:

   . 1st Method: `SSH 10 SSH 10 SSH 10 SSH 10 SSH 10 SSH 10 SSH 10 SSH 10 SSH 10 SSH 10`

   . REPEAT Method: `REPEAT 10[SSH 10]`

4. Students already familiar with the REPEAT procedure will be able to understand the programming concept that there may be more than one input inside of the brackets:

```
REPEAT 10[SSH 5 SSH 20 SSH 10]
```

6. Explain that there is yet another way of creating rhythms. This third method employs the use of the procedure SSHER:

   . 1st Method: `SSH 5 SSH 5 SSH 5 SSH 5 SSH 5`

   . REPEAT Method: `REPEAT 5[SSH 5]`

   . SSHER Method: `SSHER [5 5 5 5 5]`

7. Remind the class of their experiences with the "pizza" (circle). Review the concept that the greater the number of equal pieces, the smaller the size of each individual piece.

   . In musical terms we would say that a duration of eight is twice as long as a duration of four.

   . Experiment to find out if this is so.

```
SSHER [8 8 8 8 4 4 4 4]     (press RETURN)

SSHER [6 6 12 12 6 6 12 12]     (press RETURN)

SSHER [5 5 5 10 5 5 5 10]     (press RETURN)
```

8. In the world of music, a whole circle, or a whole note as it is more properly called, receives the longest duration of any sound.

   . Use the number 12 to represent a whole note sound.

```
SSHER [12 12 12 12 12]     (press RETURN)
```

   . Use the number 6 to represent the sound that is half (a half note) as long as the sound which the number 12 makes.

```
SSHER [6 6 6 6 6]     (press RETURN)
```

   . Use the number 3 to represent the sound that's half as long as the sound of number 6 and one quarter (a quarter note) as long as the sound of number 12.

```
SSHER [3 3 3 3 3]
```

   . Discuss how much faster the 3's sound off than the 12's.

9. Experiment with rhythms containing the whole note number 12, the half note number 6 and the quarter note number 3. Below are some examples:

```
SSHER [3 6 12 3 6 12 3 6 12 3 6 12]     (press RETURN)

SSHER [3 12 3 12 3 12 3 12 3 12]     (press RETURN)

SSHER [12 6 3 12 6 3 12 6 3 12 6 3]     (press RETURN)
```

Assignments:

1. Select nine students.

   - Three should represent whole notes, that is, having a value of 12. Another three represent half notes with a value of 6. The final three students represent quarter notes, that is, having the value of 3.

   - Allow the students to line up in front of the room in any order they choose (students should be holding signs indicating which note they represent).

   - Instruct the remaining students in the class to type in the rhythmic pattern which is created by the line.

   - Select new students and allow the process to continue, each time developing a new rhythm.

Summary:

1. We can use the REPEAT procedure in conjunction with the SSH procedure to produce rhythms.

2. We can use the SSHER procedure to produce rhythms.

3. We can use the SSHER procedure to create whole notes, half notes and quarter notes.

TOPIC 38 - A

Logo Sounds and Music

Performance Objectives - Students will be able to:

. Identify pitch using the PLAY and SING procedures.
. Use LOGO to build a C Major scale.
. Create a simple melody-line.

Vocabulary:

Motivation:

Aim: Can we use Logo to create a melody-line?

Teacher Background:

. This lesson is the last in a series of three LOGO Music and Sound lessons. Although two new LOGO procedures are introduced in this lesson, by no means should it be assumed that each procedure in itself could not be the aim of an individual lesson (or series of lessons).

Development:

1. Begin a class discussion of instruments that produce high sounds, and instruments that produce low sounds. Try to gain access to either a set of melody bells or a small xylophone.

2. Two sounds should be played for the students to hear. Challenge the students to distinguish between the higher or lower sounds when they are played. This

   MSDOS 1.00 FILE TRANSFER Ver 1.0(c) 1984 Purves Computer

   System

   - The Form:

   ```
   PLAY [pitch] [duration]
   ```

3. Pitches are represented by the numbers 0 1 2 3 4 5 6 7 8 9 10 11 12 with 0 being the lowest pitch and 12 being the highest pitch.

4. Explain that each pitch receives its own corresponding duration. Say, "Let's pair the first five pitches with a duration of 15."

   ```
   PLAY [0 1 2 3 4] [15 15 15 15 15]     (press RETURN)
   ```

. The sound just programmed seemed to be walking upstairs. Try creating a sound that seems to be walking slowly down the stairs.

```
PLAY [4 3 2 1] [30 30 30 30]     (press RETURN)
```

. Program the sound of a clock going "tick-tock."

```
PLAY [5 0 5 0 5 0] [10 10 10 10 10 10]     (press RETURN)
```

5. If we were to write all thirteen pitches, we would hear that each pitch is one half-step higher than the previous pitch. In music, this is called a chromatic scale. An octave is a scale which starts on one pitch and ends on the same pitch, only higher. An octave follows the same order as the alphabet:

0	1	2	3	4	5	6	7	8	9	10	11	12
C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C

. The symbol "#" is called a sharp. It corresponds to the black keys of a piano. All of the other pitches correspond to the white keys.

6. After some experimentation with pitches and durations, explain that there is another way of expressing what happens during the LOGO PLAY procedure. We may evoke the SING procedure to reduce the use of duration numbers:

```
1st Method:    PLAY [0 2 4 5 7 9 11 12] [20 20 20 20 20 20 20 20]

2nd Method:    SING [0 2 4 5 7 9 11 12]
```

. The duration is a constant for every pitch played when using the SING procedure.

. Rests can also be used with the PLAY or SING procedures. When one writes a rest ("R"), no sound will be heard in that slot:

```
PLAY [5 5 R 5 5] [15 15 15 15 15]     (press RETURN)

SING [5 5 R 5 5]     (press RETURN)
```

- If you want to repeat a line without having to re-type the whole line, press SHIFT/UP-CRSR.

7. Write the C Major scale. Remember, there are no sharps in this scale and one should start and end on C.

. Play a C Major Scale:

C D E F G A B C

- Now find the numbers which correspond to each letter.

  C D E F G A B C

  0 2 4 5 7 9 11 12

- Can the PLAY procedure be used with 20 as a duration? Combine the two procedures.

  ```
  PLAY [0 2 4 5 7 9 11 12] [20 20 20 20 20 20 20 20]     (press RETURN)
  ```

- Reverse the sound pattern of the scale.

  ```
  PLAY [0 2 4 0 4 0 4] [20 5 20 5 10 10 20]     (press RETURN)
  ```

8. Allow time to experiment using combinations of pitches and durations. Create long sounds as well as short sounds. For example:

   ```
   PLAY [0 2 4 0 4 0 4] [20 5 20 5 10 10 20]     (press Return)
   ```

9. Tell the class they are now going to deal with the aspect of music they are most familiar with--melody. A melody is a group of pitches which sound pleasant to the ear. For the purpose of demonstration use the familiar melody "Jingle Bells." Examine its construction:

WORDS:	Jing-	le	bells		Jing-	le	bells		Jing-le	all	the	way
PITCHES:	E	E	E		E	E	E		E	G	C	D	E
NUMBERS:	4	4	4	R	4	4	4	R	4	7	0	2	4

- Write 'Jingle Bells' using the SING procedure.

  ```
  SING [4 4 4 R 4 4 4 R 4 7 0 2 4]     (press RETURN)
  ```

- It is easier to use the SING procedure than the PLAY procedure because we are only interested in the melody line. We are not interested in modifying the duration of each note.

Assignments:

1. Write the beginning melody line to some other song. As another example you may wish to use the familiar childhood melody "Are You Sleeping":

   ```
   SING [0 2 4 0 0 2 4 0 4 5 7 R 4 5 7 R]     (press RETURN)
   ```

2. Play a version of "Name That Tune" with the class. Divide the class into two teams.

   - Have one team write tunes programming three pitches at a time. If the other team guesses the tune being played, they score 10 points. If they do not, add one additional pitch and deduct one point from 10 until the tune is finally guessed.

- Each team takes a turn with a tune. The game may either be timed, or a pre-arranged total score could determine the winner.

Summary:

1. What does pitch mean?
2. What LOGO procedure combines pitch and duration?
3. How can we reduce the use of duration numbers?

TOPIC 39 - E

Music

Performance Objectives - Students will be able to:

. Use the TOOT command in programs.

Vocabulary: TOOT

Motivation:

. Copy SOUND1, SOUND2, and SOUND3. (Note: This topic has been prepared for Atari Logo.)

```
TO SOUND1
  TOOT 0 15 15 50
END

TO SOUND2
  TOOT 0 15 15 100
END

TO SOUND3
  TOOT 0 15 15 150
END
```

. Which one is the longest lasting sound?

. Which sound is the shortest?

. Copy SOUND4. How does SOUND4 differ from SOUND3?

```
TO SOUND4
  TOOT 0 15 5 150
END
```

Aim: How can we use the TOOT command to enhance procedures?

Development:

1. Discuss the four TOOT inputs and their limits (Handout #1).

Applications:

1. Write a new program that uses TOOT in any of its subprocedures.
2. Write a program that will demonstrate:
   - . five different frequencies
   - . five different volumes
   - . five different durations

Summary:

1. How many inputs are needed with the TOOT command?
2. What are the ranges or limits of the TOOT inputs?
3. How can the TOOT command be used in creating programs?

Assignments:

1. Using SOUND1, SOUND2, and SOUND3 as models, create procedures that will:
   - . show a range of frequencies
   - . show a range of the different volumes

Handout #1:

The TOOT command needs four inputs to work. The first input tells the computer to use one of the two voices available. This input is always 0 or 1.

Example: `TOOT 0`

The second input determines the frequency. The frequency range is from 14 to beyond audibility.

Example: `TOOT 0 392`

The third input regulates the volume. The range is from 0 to 15.

Example: `TOOT 0 392 18`

The last input determines the duration of the sound made. The duration range is from 0 to 255. Each unit is equal to 1/60 second. (TOOT 0 15 15 240 lasts 4 seconds.)

Example: `TOOT 0 392 18 64`

TOPIC 39 - M

Music

Performance Objectives - Students will be able to:

. Use the TOOT command in programs.

Vocabulary: TOOT

Motivation:

. Copy SOUND1, SOUND2, and SOUND3. (Note: This topic has been prepared for Atari Logo.)

```
TO SOUND1
  TOOT 0 15 15 50
END

TO SOUND2
  TOOT 0 15 15 100
END

TO SOUND3
  TOOT 0 15 15 150
END
```

. Which one is the longest lasting sound?

. Which sound is the shortest?

. Copy SOUND4. How does SOUND4 differ from SOUND3?

```
TO SOUND4
  TOOT 0 15 5 150
END
```

Aim: How can we use the TOOT command to enhance procedures?

Development:

1. Discuss the four TOOT inputs and their limits (Handout #1).

Applications:

1. Write a new program that uses TOOT in any of its subprocedures.

2. Write a program that will demonstrate:

   . five different frequencies

   . five different volumes

   . five different durations

3. Copy NOTES.

```
TO NOTES :F
  CT
  TOOT 0 :F 15 50
  PR :F WAIT 12
  NOTES :F + 1
END
```

4. NOTES 15 demonstrates frequency ranges. It also prints the frequency.

5. How can NOTES be changed so that each sound lasts longer?

6. What will happen if the following line is added after CT?

```
IF :F > 100 [STOP]
```

7. What will happen if the recursive line is removed?

8. What will happen if in the recursive line, the one is changed to five?

Summary:

1. How many inputs are needed with the TOOT command?

2. What are the ranges or limits of the TOOT inputs?

3. How can the TOOT command be used in creating programs?

Assignments:

1. Using SOUND1, SOUND2, and SOUND3 as models, create procedures that will:

   . show a range of frequencies

   . show a range of the different volumes

Handout #1:

The TOOT command needs four inputs to work. The first input tells the computer to use one of the two voices available. This input is always 0 or 1.

Example: `TOOT 0`

The second input determines the frequency. The frequency range is from 14 to beyond audibility.

Example: `TOOT 0 392`

The third input regulates the volume. The range is from 0 to 15.

Example: `TOOT 0 392 18`

The last input determines the duration of the sound made. The duration range is from 0 to 255. Each unit is equal to 1/60 second. (TOOT 0 15 15 240 lasts 4 seconds.)

Example: `TOOT 0 392 18 64`

TOPIC 39 - A

Music

Performance Objectives - Students will be able to:

- Use the TOOT command in programs.

Vocabulary: TOOT

Motivation:

- Copy SOUND1, SOUND2, and SOUND3. (Note: This topic has been prepared for Atari Logo.)

```
TO SOUND1
  TOOT 0 15 15 50
END

TO SOUND2
  TOOT 0 15 15 100
END

TO SOUND3
  TOOT 0 15 15 150
END
```

- Which one is the longest lasting sound?

- Which sound is the shortest?

- Copy SOUND4. How does SOUND4 differ from SOUND3?

```
SOUND4
  TOOT 0 15 5 150
END
```

Aim: How can we use the TOOT command to enhance procedures?

Development:

1. Discuss the four TOOT inputs and their limits (Handout #1).

2. Copy DOWN, UP and SONG.

```
TO DOWN :HIGH :LOW
  IF :HIGH < :LOW [STOP]
  TOOT 0 :HIGH 15 4
  DOWN :HIGH * .97 :LOW
END

TO UP :LOW :HIGH
  IF :LOW > :HIGH [STOP]
  TOOT 0 :LOW 15 4
  UP :LOW * 1.025 :HIGH
END

TO SONG
  REPEAT (1 + RANDOM 4) [UP 1000 + RANDOM 50 1050 + RANDOM 100 WAIT
         RANDOM 30]
  REPEAT (1 + RANDOM 4) [DOWN 1100 + RANDOM 50 1000 + RANDOM 100 WAIT
         RANDOM 30]
  SONG
END
```

. Note: In SONG, the lines beginning with REPEAT are complete lines. Press the RETURN key after each RANDOM 30].

3. Incorporate the SONG procedure into an original program.

4. Create a program similar to SONG.

5. How can the frequencies of SONG be changed?

<u>Applications</u>:

1. Write a new program that uses TOOT in any of its subprocedures.

2. Write a program that will demonstrate:

   . five different frequencies

   . five different volumes

   . five different durations

3. Copy NOTES.

```
TO NOTES :F
  CT
  TOOT 0 :F 15 50
  PR :F WAIT 12
  NOTES :F + 1
END
```

4. NOTES 15 demonstrates frequency ranges. It also prints the frequency.

5. How can NOTES be changed so that each sound lasts longer?

6. What will happen if the following line is added after CT?

```
IF :F > 100 [STOP]
```

7. What will happen if the recrusive line is removed?

8. What will happen if in the recursive line, the one is changed to five?

Summary:

1. How many inputs are needed with the TOOT command?

2. What are the ranges or limits of the TOOT inputs?

3. How can the TOOT command be used in creating programs?

Assignments:

1. Using SOUND1, SOUND2, and SOUND3 as models, create procedures that will:

   . show a range of frequencies

   . show a range of the different volumes

Handout #1:

The TOOT command needs four inputs to work. The first input tells the computer to use one of the two voices available. This input is always 0 or 1.

Example: `TOOT 0`

The second input determines the frequency. The frequency range is from 14 to beyond audibility.

Example: `TOOT 0 392`

The third input regulates the volume. The range is from 0 to 15.

Example: `TOOT 0 392 18`

The last input determines the duration of the sound made. The duration range is from 0 to 255. Each unit is equal to 1/60 second. (TOOT 0 15 15 240 lasts 4 seconds.)

Example: `TOOT 0 392 18 64`

TOPIC 40 - E

Random Procedures

Performance Objectives - Students will be able to:

- Create random selection tools.
- Generate random numbers using Logo.
- Use random selections.

Vocabulary: random, primitive, input, ticks, wait

Motivation:

- How would you pick the winner in a class raffle?
- How does Logo make random selections?
- How could we use random to change the background color?
- How could we use random to create a spin program with random pencolors for each shape?

Aim: How can we develop and use random procedures in Logo?

Development:

1. Discuss various methods for random selection, such as names, hat, deck of cards, dice, coin, etc.

2. Discuss and demonstrate the use of random in Logo.

3. Elicit a program to randomly change the background color. The following is an example.

```
TO Color
    REPEAT 50 [SETBG RANDOM 6 WAIT 120]
END
```

4. Elicit a program to spin a square using random pen colors for each square.

```
TO Square
   REPEAT 4 [FD 40 RT 90]
END

TO Spin.square
   REPEAT 36 [square RT 10 setpc (1 + RANDOM 50]
END
```

Summary:

1. How can we generate random backgrounds and pencolors with Logo?

411

Assignments:

1. Write instructions to change the background colors quickly. Then rewrite the instructions to change them slowly.

2. Write instructions to change the background colors using only two colors...three colors.

3. Write instructions to Spin using two colors randomly...three colors randomly.

TOPIC 40 - M

Random Procedures

Performance Objectives - Students will be able to:

- Create random selection tools.
- Generate random numbers using Logo.
- Write programs using random selections.

Vocabulary: random, range, primitive, input

Motivation:

- How would you pick the winner in a class raffle?
- How are the numbers selected in state lotteries?
- How does Logo make random selections?
- How would you use Logo to run a class lottery?

Aim: How can we develop and use random procedures in Logo.

Development:

1. We will discuss various methods for random selection, such as names, hat, deck of cards, dice, coin, etc.
2. We will discuss and demonstrate the use of random in Logo. (random #, 1 + random #)
3. Write a program to spin a square using random pen colors for each square.
4. Write a program to place stars of different sizes and colors in different locations on the screen.
5. The program is:

```
TO SQUARE :SIDE
  REPEAT 4[FD :SIDE RT 90]
END

TO SPIN.SQUARE :SIDE
  REPEAT 36[SQUARE :SIDE RT 10 SETPC (1+ RANDOM 5)]
END

TO STAR :SIZE
  REPEAT 5[FD :SIZE RT 144]
END
```

```
TO SKY
  SETPC (1+ RANDOM 5)
  STAR (5+ RANDOM 15)
  PU FD RANDOM 60 RT RANDOM 360 FD RANDOM 40 PD
  SKY
END
```

Summary:

1. How can we generate random numbers using Logo and in what ways can we apply them?

Assignments:

1. Develop an algorithm to select random numbers from 10 to 20.
2. Write a program to draw concentric squares of different colors.
3. Write a program to place 100 stars at random locations on the screen.
4. Write a program that will create random drawings.
5. Write a program that will generate random sentences.

TOPIC 40 - A

Random Procedures

Performance Objectives - Students will be able to:

- Develop algorithms for generating random numbers over different ranges.
- Separate these into modules.
- Write programs using these modules to solve problems.

Vocabulary: random, range, primitive, input, initialize

Motivation:

- How would you generate random numbers for lotto?
- How would you store the numbers selected?
- How could you prevent doubles?
- How would you check for a complete game set?

Aim: How can we use random procedures as modules in programs that solve more complex problems, such as in playing lotto?

Development:

1. Write an algorithm to generate the numbers from 1 to 48.

   - The algorithm is (1 + RANDOM 48)

2. The modules are:

   - start (initialize)
   - select number
   - check for doubles
   - add to game.set
   - check for completeness

3. The program is:

```
TO LOTTO
  START
  PLAY
END

TO START
  MAKE "GAME.SET []
END
```

```
TO PLAY
  IF COUNT :GAME.SET = 6 [PR :GAME.SET STOP]
  SELECT
  IF MEMBERP :NUMBER :GAME.SET [SELECT]
  MAKE "GAME.SET SE :GAME.SET :NUMBER
  PLAY
END

TO SELECT
  MAKE "NUMBER (1+ RANDOM 48)
END
```

Summary:

1. How can we generate random values in Logo over a given range?
2. What are some of the ways we can use random generators to solve problems and to explore with?

Assignments:

1. Develop lotto for a variable upper limit.
2. Develop a random sentence generator.
3. Write an algorithm to generate random odd numbers from 1 to 99.
4. Develop an area approximator using random points. (How many fall within closed curve?)

TOPIC 41 - E
Output

Performance Objectives - Students will be able to:

- . Understand how function machines work (off the computer).
- . String together two or more function machines to create more elaborate ones (off the computer).
- . Apply the function machines on the computer through Logo.

Vocabulary: input, output, hopper, function

Motivation:

- . How would you find out how many seconds there are in a day?
- . How would you find out how many inches there are in a mile?
- . Can we develop a machine that will help us solve these types of problems?

Aim: How can we develop machines that will help us solve conversion problems?

Development:

1. What will our machine need to do to convert minutes into seconds (miles into inches)?

2. What will our machine need to do to convert hours into minutes (miles into inches)?

3. How can we string these two machines to convert hours into seconds (miles into inches)?

4. How can we develop machines (using Logo) that will help us solve conversion problems?

5. Our Logo function machines should look like this:

```
TO min.sec : min
   OP :min * 60
END

 TO hour.min : hour
    OP :hour * 60
 END

 PR min.sec hour.min #

 TO feet.inch : feet
    OP :feet * 12
 END
```

417

```
TO yds.feet :yds
   OP :yds * 3
END

TO miles.yds :miles
   OP :miles * 1760
END
```

Summary:

1. How can we develop function machines to help us solve conversion problems?

Assignments:

1. Using the function machines find out how many:

. seconds in

10 minutes
15 minutes
45 minutes
1 hour and 15 minutes
3 hours and 18 minutes
1 day

. inches in

2 feet
5 feet
2 yards
5 yards
1 mile

. feet in

2 yards
5 yards

TOPIC 41 - M

Output

Performance Objectives - Students will be able to:

- Understand how function machines work.
- String together two or more function machines to create more elaborate ones.
- Write Logo programs to create function machines.
- Use Logo function machines to solve problems.

Vocabulary: input, output, hopper, function

Motivation:

- How would you find out how many seconds there are in a day?
- How would you find out how many centimeters are in a mile?
- Can we develop machines (using Logo) that will help us solve these types of problems?
- Can you think of other problems where function machines can be used?

Aim: How can we develop machines (using Logo) that will help us solve conversion problems?

Development:

1. What will our machine need to do to convert minutes into seconds (miles into centimeters), etc.?
2. How can we translate these conversions into Logo instructions?
3. How can we write Logo programs to create these conversion tools?
4. How can we string together our Logo function machines to make more complex conversions?
5. What other kinds of problems can be solved using function machines?
6. Logo function machines will look like this:

```
TO MIN.SEC :MIN
  OP :MIN * 60
END

TO HOUR.MIN :HOUR
  OP :HOUR * 60
END
```

```
TO HOUR:SEC :HOUR
  OP MIN.SEC HOUR.MIN :HOUR
END

TO MILE.INCH :MILE
  OP FT.INCH YD.FT MILE.YD :MILE
END

TO FT.INCH :FT
  OP :FT * 12
END

TO YD.FT :YD
  OP :YD * 3
END

TO MILE.YD :MILE
  OP :MILE * 1760
END

TO INCH.CM :INCH
  OP :INCH * 2.54
END

TO MILE.CM :MILE
  OP INCH.CM MILE.INCH :MILE
END
```

Summary:

1. How can we write Logo programs to create function machines to help us solve conversion problems?

Assignments:

1. Using the function machines find out how many

   . Seconds in one month, six months, one year and three months.

   . Centimeters in 2 feet, in 2 yds, in 100 yds.

TOPIC 41 - A

Output

Performance Objectives - Students will be able to:

- Understand how function machines work.
- String together two or more function machines to create more elaborate ones.
- Write Logo programs to create function machines.
- Generate their own problems and create Logo function machines to solve them.

Vocabulary: function machines

Motivation:

- How would you find out how far light travels in a day, a month, a year, etc.?

- How would you create a process that reverses the letters of a word? the order of a list of words?

- Can we develop machines with Logo that will help us solve these types of problems?

- Can you think of other problems where function machines can be used?

Aim: How can we explore areas of problem solving where Logo function machines will be useful?

Development:

1. What will our machine need to do to convert time into light distances? (Light travels 186,000 miles per second.)

2. How can we write Logo programs to create these conversion tools?

3. What will our machine need to do to reverse the letters of a word?... the words in a list?

4. How can we string together function machines to make more complex conversions?

   - Note: not all function machines can be strung together, for example, list and word reverse machines.

5. What other kinds of problems can be solved using function machines?

6. Logo function machines will look like this:

```
TO SEC.MILE :SEC
  OP :SEC * 186000
END

TO MIN.SEC :MIN
  OP :MIN * 60
END
TO HOUR.MIN :HOUR
  OP :HOUR * 60
END

TO DAY.HOUR :DAY
  OP :DAY * 24
END

TO REVERSE.WRD :WRD
  IF EMPTYP :WRD [O "]
  OP WORD LAST :WRD REVERSE.WRD BL :WRD
END

TO REVERSE.LST :LST
  IF EMPTYP :LST [OP []]
  OP SE LAST :LST REVERSE.LST BL :LST
END
```

Summary:

1. How can explore and develop the uses of Logo based function machines?

Assignments:

1. Using the function machines find out how far light will travel in a minute, an hour, a day, a year.

2. Create a Logo machine that determines the time needed for light to travel from the sun to the earth, a distance of 93,000,000 miles.

3. Based on the reversal machines, create a Logo program to code and decode messages.

TOPIC 42 - E

Tiles and Characters

Performance Objective - Students will be able to:

- Stamp ASCII characters onto a graphics screen.

Teacher Background:

- At present only Texas Instruments LOGO (LOGO versions I & II) employ the use of "tiles" for graphics programming. In point of fact all TI LOGO Turtle graphics are drawn using tiles. Therefore, when all 256 user-definable 8-by-8-dot tiles are used up, the computer responds with the following error message: OUT OF INK
- At this point the only way to continue drawing with Turtle graphics is to clear the screen (CLEARSCREEN or CS).
- The Commodore 64 Terrapin version also offers a method of stamping characters onto a graphics screen. However, these characters are not re-definable and the procedures for this programming do not appear on the C-64 LOGO boot disk. Teachers interested in this programming technique for the C-64 LOGO should refer to the C-64 LOGO Utilities Disk and examine the "STAMPER" and "STAMPFD" files.
- In both the TI and C-64 versions the full range of uppercase ASCII characters are available for use on the graphics screen.

Vocabulary:

Motivation:

- After distributing a copy of your computer's "Monitor Grid" to each student, examine the pattern of rows and columns.
- Tell the students that, although the same exact pattern of rows and columns also exists on the LOGO monitor screen, they cannot see the boxes because they are "invisible."
- The student's "job" is to locate the invisible boxes and then place characters inside them.

Aim: Can we print messages onto the graphics screen?

423

Development:

1. Type the following into the computer:

```
PC 72 PC 73     (press ENTER)
```

   - . What do you see? (The word HI) Referring to the sheet, explain that each box is called a "tile" and we just placed an "H" and an "I" into the first two tiles on the monitor screen using the PRINTCHAR (PC) primitive command.

2. All characters, which include the letters and numbers, are assigned a numerical value by the computer. Distribute the ASCII Character Code Sheet.

   - . Optional: Discuss the origin and meaning of the ASCII code.
   - . For younger students, it is often advantageous to describe numerical values as secret codes. From this perspective words and sentences may become part of a game to decipher secret messages.

3. Refer to the ASCII Character Code Sheet and review the 64 commonly used characters.

   - . Using a street address as an analogy, lead the class to the conclusion that by finding the column and row number of a tile we can identify its exact location (address).

4. Select specific tiles on the Monitor Grid Sheet. Challenge the students to identify the addresses of these tiles.

   - . To facilitate the presentation, the Grid Sheet may be transfered to an overhead acetate.

5. How can we direct the computer to place a specific character into a specific tile?

   - . Describe the primitive procedure PUTTILE (PT) and its programming application:

```
PUTTILE  #  #  #
```

   - . The first # prints the character with that particular numerical code, the second # is the column address, and the last # is the row address.
   - . Write the following on the board:

```
PUTTILE  90  16  12
```

     - Challenge the group to identify the character and location on the Monitor Grid.

Assignments:

1. Write the alphabet across the monitor using the PUTTILE command.

2. Use the turtle to draw a box (square). Then stamp the word "box" ("square") inside of the drawing.

   . Repeat this procedure for a rectangle and a triangle.

3. Use the PUTTILE command to describe, in sentences, an original turtle or sprite procedure.

Summary:

1. What is the ASCII code?

2. What are tiles and how many are there?

3. How can a tile be located?

4. What does the primitive command PUTTILE # # # do?

TI ASCII CHARACTER CODE CHART

GROUP 1

CODE NUMBER	CHARACTER
32	(SPACE)
33	!
34	"
35	#
36	$
37	%
38	&
39	'

GROUP 2

CODE NUMBER	CHARACTER
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/

GROUP 3

48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7

GROUP 4

56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?

GROUP 5

64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G

GROUP 6

72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O

GROUP 7

80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
	T

GROUP 8

88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
92	

TOPIC 42 - M

Tiles and Characters

Performance Objective - Students will be able to:

- Use tiles and characters to create designs.

Motivation:

- There is another use for stamping characters onto a monitor screen. The skills learned in the previous lesson can now be used as a third method for making graphic designs in LOGO:

  - Method #1: Turtle Mode
  - Method #2: Sprite Mode
  - Method #3: Tiles and Characters

- By using the third method, students will be able to use all the characters on the keyboard, in addition to turtle and sprites, when they create a LOGO world."

Aim: Can we create designs with tiles and characters?

Development:

1. Distribute a copy of the "Monitor Grid" and/or use an acetate version on an overhead projector.

2. Draw an outline of the letter "L."

   - If you are using an overhead projector (preferred method), demonstrate by outlining an "L" somewhere in the middle of the grid.

     - Ask the students to try and copy the exact size and location of the "L" so that they may have a duplicate in front of them.

     - Be certain that no tiles are filled in at this time.

   - Who can identify the column numbers and row numbers of the uppermost tiles in this "L"?

   - After several tile coordinates have been offered, explain that there is a much easier way to organize the information coming in.

3. Describe and set up a tile plan:

TILE PLAN

NAME:____________________

CHARACTER CODE NUMBER	COLUMN NUMBER	ROW NUMBER

4. If the students are familiar with cassette or disk file retrieval, coordinate the tile plan name with the file name on the cassette or disk.

5. Continue recording tile locations, placing all coordinates into the tile plan.

6. What is the character code for "Z"?
   - How could we build our 'L' out of Zs? Fill in the remainder of the tile plan to accomplish this.
   - Remind the students they are merely making a large "L" out of smaller Zs.

7. Challenge the students to enter their tile plan into the computer.

Assignments:

1. Write a tile plan that draws the student's initials. Select any character to draw with.

2. Draw "LOGO" onto the screen using each letter's character as the design.

Assignments for Advanced Students:

1. Add a character border to any turtle or sprite procedure previously programmed. Identify and label individual items in the picture or design (e.g. "window", "tree", etc.).

Summary:

1. How can tiles and characters be used to create designs?

2. How can a tile plan help organize tile designs?

3. How can we identify or label individual items in a picture or design (e.g. "window," "tree," etc.)?

TOPIC 42 - A

Tiles and Characters

Performance Objectives - Students will be able to:

- Integrate colors into tile and character designs.

Teacher Background:

- All TI LOGO tiles have two colors--a background color and a foreground color. Tile colors coincide with turtle and sprite colors; all colors may be accessed using the same color code numbers (see attached sheet).
- TI ASCII characters are grouped in sets of eight (see attached sheet). If one alters the color of any individual member in a character set, all characters in that set also change color.

Motivation:

- Tell the students that the full range of LOGO colors are available to them when they program with tiles and characters.
- Say, "Instead of the usual black lettering and character generation, you can write messages or create character designs with any combination of LOGO colors you desire."

Aim: Can we create color designs using tiles and characters?

Development:

1. Review the following:

   - There are 256 tiles on the monitor screen.
   - There are 16 LOGO colors.
   - The LOGO monitor has 31 columns and 23 rows.

2. How do we tell the turtle or sprite to change color? Discuss the answers:

```
TELL TURTLE                    OR                  TELL SPRITE  #
SETCOLOR :color                                    SETCOLOR :color
```

   - Convey the idea that we can "talk" to tiles the same way they "talk" to the turtle or sprites.

3. Enter the following character into the column and row identified.

```
PUTTILE 35 16 12     (press ENTER)
```

4. Tell Tile #35 to become red.

```
TELL TILE 35
SETCOLOR :RED
```

5. Distribute a copy of the Character Set Groups sheet to each student. Examine the way in which characters have been organized into groups of eight.

   . Enter the following characters into tiles somewhere near the center of the monitor. The exact location is not important.

   ```
   H I J K L M N O
   ```

   . When the students have successfully programmed these characters, ask them to type in the following:

   ```
   TELL TILE 72
   SETCOLOR :BLUE
   ```

   . Before pressing ENTER ask the students to predict the results.

6. What portion of the tile is changing color?

   . Only the background has changed; the character in the foreground remains unchanged. Explain that LOGO provides a procedure for changing the colors of both the foreground and background simultaneously.

7. Write the form:

   ```
   TELL TILE #
   SETCOLOR [#  #]
   ```

   . The first "#" within the SETCOLOR brackets is the color code number for the foreground.

   . The second "#" within the SETCOLOR brackets is the color code number for the background.

   . Describe the effect of the following program:

   ```
   TELL TILE 72
   SETCOLOR [15  4]     (press ENTER)
   ```

   . Predict the effects of the following.

   ```
   REPEAT 100[PRINTCHAR 72]     (press ENTER)
   ```

Assigments:

1. Divide the monitor screen into colored stripes (HINT: use character #32).
2. Use blue tiles to color an ocean, and sprites to construct a sailing ship that moves through the water.
3. Use the turtle to draw a restaurant. Use characters and tiles to construct a flashing sign atop the building.

Summary:

1. Which LOGO procedure may be used to color tiles?
2. How can we simultaneously color the foreground and background of tiles using the LOGO procedure SETCOLOR?

TI LOGO COLOR CODE

COLOR	NUMBER
CLEAR	0
BLACK	1
GREEN	2
LIME	3
BLUE	4
SKY	5
RED	6
CYAN	7
RUST	8
ORANGE	9
YELLOW	10
LEMON	11
OLIVE	12
PURPLE	13
GRAY	14
WHITE	15
OLIVE	12
PURPLE	13
GRAY	14

TOPIC 43

Advanced Topic: Passing Values

Performance Objectives - The student will be able to:

- Explain how the same variable can be shared by several procedures.
- Use subprocedures which receive values from other procedures.

Vocabulary: Variables, procedure, passing by value.

Motivation:

- How can two different procedures use the same variable name?

  - One procedure can take a "copy" of the variable and use it or change it.

Aim: To develop an understanding of passing by value in a Logo program.

Development:

1. Consider the following procedure

```
TO LEAF :L
  IF :L>60 THEN LEAFB :L
  IF :L=61 THEN STOP
  PU FD :L PD FD :L PU BK :L * 2
  RT 1
  LEAF :L+1
END
```

2. The procedure LEAF draws the upper portion of the leaf below. Notice that the variable L is 61 when the upper portion of the leaf has been drawn.

   . Then the lower portion of the leaf is drawn by LEAFB.

```
TO LEAFB :L
  IF :L < 0 THEN STOP
  PU FD :L PD FD :L PU BK :L * 2
  RT 1
  LEAFB :L-1
END
```

3. When LEAFB is called, the variable L is 61. LEAFB reduces L for each recursion. The final value of L in LEAFB is -1. The LEAF procedure will now end.

4. Does the LEAF procedure end with the value L = -1 or L = 61? (L = 61)

Summary:

- The changes of value to L in LEAFB have no effect on the value of L in the calling procedure LEAF.
- LEAFB receives a copy of the value passed to it from the calling procedure. This preserves the final value of L in the calling procedure.

Assignments:

1. Vary L and observe the effects on the shape of the leaf.
    - What is the value of L at the beginning of LEAFB, at the end of LEAFB, and at the end of LEAF?

TOPIC 44

Advanced Topic: Approaching Infinity

Performance Objectives - The student will be able to:

- Approximate a circle by rotating a square whose diagonal is the radius of the circle.
- Define the process of getting larger and larger as approaching infinity.
- Recognize that as the number of squares increases to infinity the figure comes closer and closer to a true circle.

Vocabulary: Infinity, inscribed.

Motivation:

Aim: To develop an understanding of approaching infinity by examining a Logo program that illustrates that concept.

Development:

- Define the SQUARE procedure below.

```
TO SQUARE
  REPEAT 4  FD 90 RT 90
END
```

- What will happen if we repeat SQUARE while rotating it slightly?

- Try the following procedure with different inputs.

```
TO CIRCLE :S
  REPEAT :S  SQUARE RT 360/:S
END
```

- What happens as the variable S gets larger and larger (S approaches infinity)?

Summary:

- As S approaches infinity the figure generated approaches a circle.

- The diagonal of the square is the radius of the circle.

Projects:

- Generate a circle using triangles, hexagons, etc.

APPENDICES

Appendix A
Computers and Special Education Students

Special educators must prepare students to live in a time when many aspects of community life will be computer-dependent. Daily living skills, such as banking, using the library and shopping, are rapidly undergoing computerization. Skills taught only a few years ago are, in some cases, already obsolete. Special educators face the challenge of preparing their students to live in a world in which computer literacy may well prove to be as important as book literacy.

Our present special education program attempts to serve students with a wide range of intellectual, motor and communications skills. In recent years there have been tremendous advances in the range and sophistication of adaptive devices to be used with microcomputers. Such devices aid those with special needs in their use of computers.

For the blind, there are Braille keyboards, Braille printers, and programs that convert Braille to English and vice versa. The visually impaired can make use of a device that enlarges the on-screen letters up to sixteen times their normal size.

Software is now available which helps teach lip-reading through animated graphics. Another program depicts the pitch and frequency of the teacher and speech student's voices side by side, and learning can take place using a very accurate feedback method.

A number of control devices have been developed for the physically disabled. These devices allow even individuals with only one means of physically controlling their environment to interface with the microcomputer. For example, the individual with good head control can use an optical headpointer; a manual pointer can be utilized by someone with fine motor skills; those with more erratic control can use a wobble switch; for quadriplegics with speech ability, there are voice-activated systems.

By maneuvering one switch in any of the above fashions, the user can perform quite sophisticated functions. It is becoming possible to operate the computer with more subtle body cues. Handicapped students who were formerly excluded from the mainstream of education may no longer face obstructions to learning.

There are a great number of ways in which the microcomputer can be utilized in the special education classroom. The computer can be utilized as a type of "electronic chalkboard." The teacher inputs data which could be viewed by the entire class on a large monitor. The teacher might also call on individual students to input a line of data while the rest of the class looks on. In this way, the computer is utilized as a tool for class participation.

This type of lesson would best be followed by a reinforcement in which students went to separate terminals. The smaller class size of most special education classes makes this economically feasible.

There are several important reasons why a computer curriculum should be utilized by special education. The computer allows students to make errors in a non-threatening way. It has infinite patience (if so programmed) and is non-judgmental in its feedback. This form of non-threatening feedback may help decrease the great amount of stress which is so evident in many of our special education classrooms. Some research indicates that many learning-disabled students show a decline in impulsive behavior while working at the computer and a concommitant decline of impulsive behavior in other learning activities.

Many of our special education students have limited success with paper and pen activities. The highly motivating and non-threatening aspects of the computer may allow us to reach a few new students. The individualized nature of computer instruction will allow the teacher to spend a greater amount of time working with students on a one-to-one basis. At the same time, individualized software assures that each student at a terminal is receiving relevant work and immediate feedback. Careful selection of software packages will assure that the needs of both the fast and slow learner are met. The computer will allow each student to work at his/her own pace.

Many special education students display a greater need for repetition and routine than their regular classroom counterparts. Drill and practice, which is often tedious, can be made more exciting through the use of animation, sound effects and game-playing situations. Thus, the need of many of our students for "overlearning" can be met without the usual dread.

Finally, the problem-solving aspects of computer programming allows special students to move out of the role of passive learner into the role of active problem solver.

The educational potential for the computer in the classroom is great, but there are some problems which educators have to deal with. Most computer assisted instruction is transmitted visually. Special education students with reading problems may have difficulty following the written instructions. Therefore, teachers should be circulating around the room and reinforcing the visual cues with verbal directions.

Another potential problem is that a computer-delivered tutorial system, may, by gaining in ability to individualize instruction, lose in teaching social interpersonal skills. Many special educational students have a need to develop more positive social skills. Will the computer kill the chance to develop these skills in the classroom? Again, the teacher can make a big difference here. By adding human reinforcement in the form of praise or an approving touch, we can keep the student from becoming too isolated. Teachers may want to allow two or more students to work together on a particular program. This should not be offered as a rationale for having more students per computer. The wide variety of learning styles and individual needs among our special population calls for one child per computer wherever possible. Since the special education class is usually smaller in size than the mainstream class, this is generally not a problem.

A third potential problem is the possibility of using the computer to instruct in areas that could best be presented in another medium. One should not try to teach skills which are best presented auditorily with a computer which relies so heavily on visual material. Advances in the development of the talking computer and other technological innovations will allow teachers to utilize computers in teaching to a student's strongest modality.

Yet another area of concern is the possible use of the computer as an "electronic babysitter." There are many arcade-type games available for use with micros. While these games can serve as reinforcers in certain instances, one must be careful not to overuse them.

Finally, it seems that the technological innovations in computer hardware have outdistanced the development of appropriate software in the special education field. There is a great challenge for special education teachers to develop software which will serve not only as electronic textbooks but will maximize their students' potential.

In general, software systems designed for special education should include the following characteristics:

1. It should have diagnostic and prescriptive strategies which place the user at the correct level.

2. It should be organized into a hierarchy of skills.

3. It should have clear and measureable performance objectives.

4. It should be individualized and provide immediate feedback to user responses.

5. It should utilize a multi-sensory approach.

Teachers who have little experience in programming can still be of great help in developing good computer instruction programs. First, it is important for them to preview hardware that is going to be purchased by their schools. Will it meet the needs of the special students in their classrooms? Furthermore, teachers need to review and constructively criticize computer software in much the same way they evaluate other curricular material.

Appendix B - Glossary

Array An array is a collection of items that are all labeled with the same *variable* but have different *subscripts.* A(1), A(2), A(3), A(4), and A(5) make up an array of variables; they are all named A, but A(1) is the first A, A(2) is the second A, and so on. An expression such as A(E), where E is being used to count errors (for example), is also an array, because the value of E will increase with each *iteration.* So A(E) will be, successively, A(0), A(1), A(2), etc.

Binary Binary numbers are based on the number 2, instead of the usual base of 10. Binary place values are 1, 2, 4, 8, 16, 32, etc., instead of the more familiar 1, 10, 100, 1000, etc. With this system all numbers can be written with two symbols, 0 and 1, instead of the usual ten sumbols (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). All typewriter characters can be translated into binary, via the ASCII code. Computers perform all manipulations, both of words and of numbers, in binary arithmetic.

Bit A bit is a single digit in the *binary* code. Both numbers and letters are encoded as BInary digiTs—hence, bits.

Bug A bug is a defect. This word generally refers to an incorrect statement or sequence of statements in a computer *program.* (Chapter 1)

Byte The entire *binary* representation of a character is called a byte. In the computer's binary code the letter *A* looks like 1000001, *B* looks like 1000010, and so on. The whole expression is a byte; each individual digit is a *bit.*

CAI Computer-assisted instruction uses the computer for drill or practice of specific, factual information. Essentially, CAI employs a computer as an electronic workbook.

Compiler A compiler is a *program* that converts a more human-like computer language, such as BASIC or COBOL, into the computer's own *machine language.* A compiler converts an entire program into machine code all at once and the user can then either save or run the resulting machine code.

CRT (Cathode-ray tube) This is a standard television picture tube, the most common display device used with microcomputers.

Cursor The cursor is a visual character that a computer prints on its screen to show where the next piece of information will appear. As you type, your letters are displayed on the screen, and the cursor moves along to show you where your next input will be displayed.

Data Data is information that a computer holds or manages: for example, a mailing list or a *program.* Any information the user asks a machine to remember is called data.

File A file is a stored collection of *data.* Computers can store information either internally in memory or on external devices.

Graphics Graphics is the visual material displayed on the *CRT* screen or output on paper. Cartoon characters, pencil drawings, and even ordinary typewriter characters are forms of computer graphics.

Integrated circuit chip The invention of the integrated circuit chip set off the whole microcomputer revolution. The chips themselves are small plastic, metal, and silicon devices that contain thousands of separate storage locations. They have replaced the tubes and transistors used in earlier machines, since they are both faster and more durable. A single silicon chip about one inch long can store many thousands of pieces of information. There are chips now in production that can each hold more than a million facts.

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

Interface An interface is a piece of hardware used to connect two devices, to translate between them. Each manages *data* in a different way. A printer, for example, moves data very slowly but a CPU moves data much more quickly. An interface allows the two to exchange data by taking the data in at one speed and sending it out at another.

Interpreter Like a *compiler*, an interpreter is a *program* that translates the more human-like computer languages into the machine's code. Interpreters are slower than compilers, since they translate each instruction every time it is used. Interpreters usually require much less memory space than compilers do.

I/O devices Input/output devices are pieces of hardware that route information into and out of the CPU. Disk drives can serve both input and output functions, but the keyboard and *CRT* screen are unidirectional I/O devices.

Iteration Iteration involves counting. Each time the computer repeats the series of steps that form a given operation constitutes one iteration.

K K stands for "thousands." When a computer company says that their machine has 48K of memory capacity, they are telling you that the machine can hold approximately 48,000 characters (or *bytes*) in memory at one time. (Actually 1K = 1024 *not* 1000, but 1000 is close enough for our purposes.)

Machine language Machine language is the code that computers actually use to perform their operations. It consists of commands in the form of binary numbers that cause the computer's electronic switches to open and close.

Modem The MOdulator/DEModulator converts information from computer-readable *binary* digits into sound and then back again. It is used to interface a computer to a telephone line, so that one computer can talk to another a great distance away.

Monitor program The monitor program comes in the *ROM* of the computer as purchased. It directs the machine's response to keyboard strokes and its transmission of *data* to output devices. It thus provides the CPU with orderly connections to its input and output devices.

Numeric Numeric means "relating to numbers." A numeric *variable* is a label that is affixed to a number when it is stored in computer memory.

Peripherals Parts of a computer that work in connection with it but are not part of it. Devices like tape recorders, disk drives, printers, microphones, and speakers are often used as peripheral equipment.

Program A collection of *statements* that tells a computer how to do a specific job.

Program documentation Program documentation is material that clarifies the nature of a *program*. It comes in two forms: External documentation is the printed matter—manuals, diagrams, and so on—that describes the parameters of a program and, essentially, guides the user through the required interactions. Internal documentation is actually part of the program. Within this category there are two subgroups: user documentation and programmer documentation. The first consists of the PRINT *statements* that tell the user what to expect as the program flows through its course. The second comprises the REMark statements used by the programmer to label different sections of the program.

RAM Random access memory is the space where the computer stores *programs* and other *data*. Material in RAM can be erased and is accessible to programmers.

ROM Read-only memory is information that is stored in a computer and cannot easily be altered. *Compilers* and *interpreters* are often stored in ROM, which is also called "hardwired software" or "firmware."

Simulation A *program* that allows a user to test solutions to "real life" problems. The user suggests an answer to the problem and the computer projects the consequences. Simulation allows users to test out different approaches to a situation on a machine, without the risks and difficulties of experimenting in the real world.

Software Software refers to sets of instructions with which a programmer causes the computer to perform specific tasks.

Statement A statement is a computer instruction of one or more words that tells the computer to do a specific task. A sequence of statements forms a *program*.

String A string is a series of keyboard characters—a word or a collection of words and numbers—that makes up a piece of information. Street addresses or even whole sentences or paragraphs can be strings.

String variable A string variable is the label that is affixed to a *string* when it is stored in memory. The computer programmer can access such stored information by using the appropriate variable name.

Subroutine A subroutine is a portion of a program that a programmer needs to access repeatedly. Instead of writing these instructions over each time they are needed, the programmer can put them in a special place in the program and ask the computer to go there, implement the subroutine *statements*, and then return to where it left off (its "calling spot") in the *program*. This technique greatly reduces the size of many programs.

Subscripted variable This is another name for an *array* variable.

User A user is a person who uses a computer. Computer users, including those in the field of education, often know very little about computers, but they do know how to follow directions. They simply turn on their machines and follow the English instructions provided by a prepackaged *program*.

Variable A variable is a label affixed to words or numbers that you wish to store in the computer. The computer remembers not only the *data* but also the variable that identifies them, so that you can retrieve or change the information later.

Word processing Writing documents with the aid of a computer. We wrote this book with a computer and a word-processing *program*. It enabled us to type in the text, modify it, store it, and print it out in a formatted (i.e., standardized) fashion. Word processing reduces the time spent typing repetitive documents; it can cut in half the time spent on secretarial functions.